Популярные лекции
по математике

Н. А. АРХАНГЕЛЬСКИЙ,
Б. И. ЗАЙЦЕВ

АВТОМАТИЧЕСКИЕ ЦИФРОВЫЕ МАШИНЫ

*

ГОСУДАРСТВЕННОЕ ИЗДАТЕЛЬСТВО
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА · 1958

ПОПУЛЯРНЫЕ ЛЕКЦИИ ПО МАТЕМАТИКЕ
ВЫПУСК 28

Н. А. АРХАНГЕЛЬСКИЙ и Б. И. ЗАЙЦЕВ

АВТОМАТИЧЕСКИЕ ЦИФРОВЫЕ МАШИНЫ

6886

Б[illegible]
МАТ[illegible]

ГОСУДАРСТВЕННОЕ ИЗДАТЕЛЬСТВО
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
МОСКВА 1958

11-2-1

АННОТАЦИЯ

Брошюра в доступной форме знакомит читателя с новой быстродействующей вычислительной техникой и с перспективами ее использования для решения труднейших научных задач, для автоматизации управления производственными и другими процессами, для облегчения и ускорения самых различных видов умственного труда. Описываются основы устройства электронных цифровых машин и дается понятие о подготовке программ для машинного решения различных задач. Книга иллюстрирована большим количеством рисунков и фотографий. Рассчитана на школьников старших классов и на широкие круги интеллигенции.

ОТ АВТОРОВ

Перед авторами была поставлена задача в небольшой книжке рассказать о принципах работы электронных цифровых вычислительных машин и познакомить читателя с основами программирования. По замыслу составителей излагаемый материал должен быть доступным для читателей, оканчивающих средние учебные заведения. Для понимания ряда глав и упражнений от читателя, кроме желания, потребуется и некоторое напряжение мысли. В главах, затрагивающих техническую сторону вопроса, авторы умышленно, в целях простоты, не приводили принципиальных схем устройств.

Авторы выражают глубокую благодарность проф. А. А. Ляпунову за ценные советы, данные при просмотре рукописи.

Н. А. Архангельский, Б. И. Зайцев

ВВЕДЕНИЕ

Привычное представление о машине, как о некотором грохочущем сооружении из шестеренок и рычагов, рассеивается, когда вы входите в зал, где установлена электронная вычислительная машина «Стрела». Тишина здесь нарушается лишь равномерным шелестом струй воздуха, охлаждающих детали машины.

Три большие стойки, сверкающие красноватыми огнями нескольких тысяч неоновых ламп, и пульт управления с множеством кнопок и переключателей создают необыкновенную картину, поражающую воображение человека. Эта сложнейшая установка содержит свыше 5000 электронных ламп, 40 000 полупроводниковых диодов и около 100 000 других радиодеталей. «Стрела» — одна из многих вычислительных машин, созданных советскими учеными и инженерами.

Для каких же целей нужны подобные машины?

Ответим на этот вопрос следующим примером. Пусть для решения задачи с несколькими неизвестными нужно сначала составить систему уравнений. Для этого иногда приходится приложить немало усилий. От человека, решающего задачу, часто требуется не только отличное знание математики. Чтобы правильно составить систему уравнений, нужно, как правило, еще и понимать физический смысл задачи, нужно уметь сопоставить различные физические явления и учесть из них только те, которые могут ощутимо повлиять на искомые ответы, нужно найти такие методы решения задачи, чтобы они были простыми, давали точные результаты и быстрее приводили к цели. Другими словами, этот первый этап работы, который мы называем творческим, требует от человека наибольшего напряжения его умственных способностей и использования ранее накопленного опыта. А дальше?

Для решения полученной системы уравнений дальше нужно вместо букв, входящих в формулы, подставить числовые значения,

а затем терпеливо и аккуратно выполнить некоторые арифметические действия. Хорошо, если исходные формулы просты и таких арифметических действий наберется немного. Но как быть, если задача очень сложная и для ее решения нужно выполнить десятки или сотни миллионов арифметических действий? Число арифметических действий, требуемых для решения задач современной науки и техники (предсказание погоды, расчеты орбиты искусственного спутника), именно такого порядка. Для этого человеку нужно работать в течение месяцев или лет, однообразно складывая, вычитая или умножая числа. Этот второй этап работы уже не требует никаких творческих способностей от человека и, как оказывается, может выполняться машиной.

Однако вычислительные машины, которые существовали до электронных машин, работали настолько медленно, что для решения отдельных задач нужно было бы тратить по меньшей мере столетия. Ясно, что при таком темпе вычислений полученные ответы оказывались бы не актуальными.

Итак, для того чтобы освободить человека от механической вычислительной работы и ускорить решение задач, чтобы не потерять актуальности результатов, и были созданы электронные вычислительные машины.

Как будет видно дальше, эта область применения электронных вычислительных машин оказалась первой, но далеко не последней.

I. ОБЩИЕ СВЕДЕНИЯ О ВЫЧИСЛИТЕЛЬНЫХ МАШИНАХ

1. Типы вычислительных машин. Области применения

Вычислительные машины можно разделить на две группы: машины непрерывного действия и машины цифрового действия. Машины непрерывного действия оперируют с физическими величинами, изображающими числа. К таким величинам могут относиться электрическое напряжение, скорость вращения, угол поворота оси и т. д. Машины конструируются с таким расчетом, чтобы взаимодействие их частей и процессов, протекающих в них, в совокупности обеспечивало бы решение той или иной задачи.

Можно условиться, например, что в машине каждый вольт напряжения будет соответствовать числу — единице. Тогда число 62,5 будет изображено напряжением 62,5 *в*. Один градус угла может изображать единицу, тогда угол 16° будет изображать число 16, и т. д. Исходные данные задачи и результаты решения также изображаются физическими величинами. Так, чтобы узнать результат вычислений, измеряют напряжение в определенном месте машины и по нему находят число.

Примером вычислительного прибора непрерывного действия может служить экспонометр с фотоэлементом, применяемый для определения экспозиции и диафрагмы при фотографировании (рис. 1). Здесь световой поток от объекта съемки преобразуется фотоэлементом в электрический ток, который отклоняет рамку и стрелку гальванометра. Угол поворота оси рамки показывает степень освещенности объекта. Отсчитывая угол в делениях шкалы, мы тем самым получаем значение освещенности объекта, заранее пересчитанное в некоторое условное число. Используя это число, уже легко получить время экспозиции и диафрагму.

Вычислительными приборами непрерывного действия являются и такие приборы, как вольтметр, который находит разность потенциалов ($U_2 - U_1$) между двумя точками, ваттметр, который находит произведение $I \cdot U$, счетчик энергии, который находит произведение $I \cdot U \cdot t$, и др.

В машинах непрерывного действия точность вычислений находится в прямой зависимости от точности изготовления и взаимодействия применяемых деталей, от физических и механических процессов. Обращаясь снова к примеру с экспонометром, легко догадаться, что на точность его показаний может влиять и качество фоточувствительного элемента, и конструкция опорной системы осей рамки, и тщательность нанесения делений шкалы и, наконец, субъективность метода отсчета.

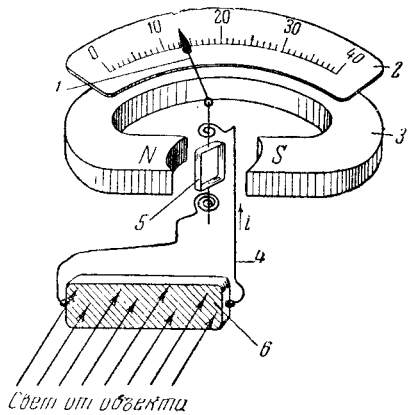


Рис. 1. Схема фотоэкспонометра:
1 — стрелка прибора; *2* — шкала показаний прибора; *3* — постоянный магнит; *4* — ток фотоэлемента; *5* — рамка измерительного прибора; *6* — фотоэлемент.

При ряде недостатков машины непрерывного действия оказываются незаменимыми в тех случаях, когда круг задач ограничен, когда требуются большие скорости вычислений и когда не преследуется цель достижения большой точности.

В отличие от машин непрерывного действия в цифровых машинах числа изображаются дискретными физическими процессами, или дискретным рядом значений, принимаемых величиной, характеризующей непрерывный процесс. Примерами цифровых машин, устройств или их элементов могут служить счеты, арифмометр, электромагнитное реле и даже обыкновенная электрическая цепь с кнопкой и батареей на одном конце и со звонком на другом.

Во всех приведенных примерах мы непременно заметим одну общую особенность. Костяшки счетов при вычислениях передвигаются в одно из крайних положений. Если же костяшка окажется в промежуточном положении, то трудно сказать, какая цифра набрана в данном разряде числа. Вероятнее все-

го, что это сделано ошибочно. Набирая число на арифмометре, мы обязательно должны устанавливать рычажки против цифр, а не между ними. Реле может находиться только в двух крайних состояниях — якорь притянут или нет. И, наконец, в последнем примере, нажимая кнопку или отпуская ее, мы заставляем звонок звонить или молчать, среднего состояния быть не может. Разумеется, во всех примерах имеется в виду, что устройства заведомо исправны. Совершенно очевидно, что с помощью кнопки и звонка можно передать любое целое число на расстояние. Для этого кнопку нужно нажать столько раз, сколько единичек содержит число. Количество звонков, подсчитанных на другом конце цепи, покажет, какое было передано число. Подчеркнем, что мы интересуемся только числом звонков и не обращаем внимания на силу, длительность и другие физические характеристики звука.

Хотя для передачи числа были использованы те же физические величины, что и в непрерывных машинах, существенно отличным оказался способ изображения числа, который называется цифровым, или дискретным.

Этот факт приводит к очень важной особенности цифровых машин, заключающейся в том, что точность их работы зависит только от количества разрядов, предусмотренных для изображения чисел, и практически не зависит (в широких пределах) от физических и геометрических свойств применяемых деталей и количественных характеристик физических и механических процессов, происходящих в машине. Иллюстрацией этого утверждения могут служить те же счеты, где точность вычислений явно не зависит от размеров костяшек, или цепь со звонком, где напряжение батарей и сопротивление проводов могут влиять лишь на несущественную для счета единиц силу звука.

Цифровые машины делятся по назначению на универсальные и специальные. Первые предназначены для решения весьма широкого круга задач из различных областей науки и техники и, как правило, бывают громоздкими и дорогими. Вторые, более распространенные, предназначены для решения одного или нескольких типов задач. Несмотря на то, что специализированные машины обладают более ограниченными возможностями по сравнению с универсальными, они получили большее распространение из-за экономичности и простоты в обслуживании.

В особые группы нужно выделить управляющие цифровые машины, предназначенные для управления производственными процессами (поточные автоматические линии, станки с программным управлением, управление химическими процессами и т. д.),

и машины для решения логических задач (перевод с одного языка на другой, задачи моделирования биологических процессов и др.). Хотя подобные машины и обладают функциями счета, результатом их работы являются не числа, а управляющие сигналы для каких-либо механизмов или логические выводы.

В истории развития цифровых вычислительных машин заслуживают внимания работы английского математика Чарльза Беббиджа. Более 100 лет назад Беббидж предложил проект «Дифференциальной машины» для вычисления математических таблиц. Второй проект Беббиджа предполагал постройку «Аналитической машины», предвосхитившей своими принципами современные цифровые машины. По идее Беббиджа «Аналитическая машина» должна была обладать следующими особенностями: выполнять простые арифметические действия; запоминать начальные данные, промежуточные данные и результаты вычислений; запоминать группу инструкций или команд, по которым должно выполняться решение задачи; выдавать результаты вычислений и последовательно выполнять команды заданной программы вычислений. Особый интерес представляла заложенная в машину способность автоматически прекращать вычисления при выполнении некоторых условий. Если такое условие не было достигнуто при первом выполнении программы, то цикл вычислений повторялся. На рис. 2 изображена часть «Аналитической машины» Беббиджа.

Однако и до Беббиджа было много попыток строить цифровые вычислительные машины. Так, например, в 1642 году французский математик Паскаль демонстрировал изобретенную им суммирующую машину. В 1694 году немецкий математик Лейбниц демонстрировал суммирующую и множительную машину. В 1855 году швед Георг Шутц использовал основные идеи Беббиджа и построил большую цифровую машину, пригодную для практического использования. В 1874 году инженер Однер построил арифмометр, который явился прототипом современного арифмометра.

Широко известны работы великого русского математика П. Л. Чебышева (1821—1894), которому принадлежат классические открытия в теории чисел, теории вероятностей, в теории механизмов. В 1878 году Чебышев сконструировал арифмометр, отличительная особенность которого заключалась в весьма оригинальном приспособлении для переноса десятков из младших разрядов в старшие. Арифмометр Чебышева считался наиболее совершенной из существовавших тогда машин

этого рода. На рис. 3 приведен общий вид арифмометра Чебышева.

В 1909 году испанец Людгейт построил цифровую машину, которая могла выполнять все арифметические действия и печатать результаты на бумаге. Однако, как и все предыдущие, за исключением машины Беббиджа, машина Людгейта не имела элементов программирования вычислений.

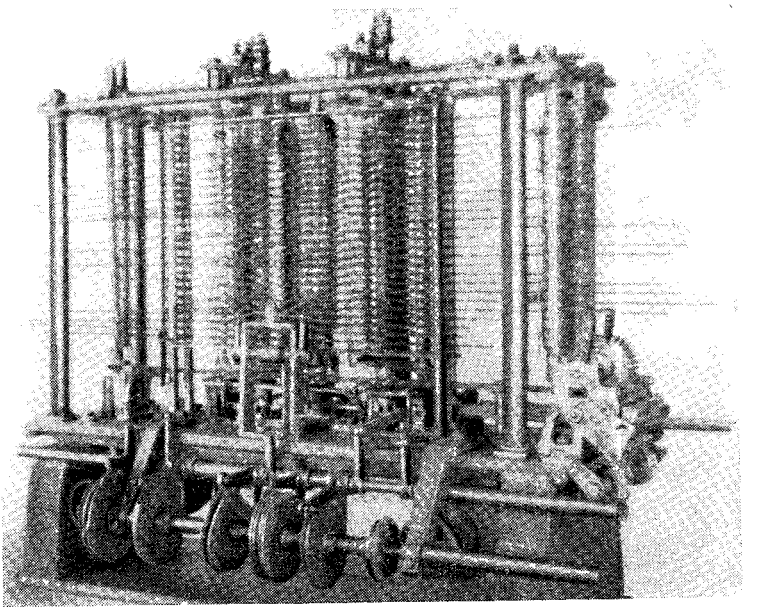


Рис. 2. Часть «Аналитической машины Беббиджа».

До 1937 года цифровая вычислительная техника развивалась главным образом в направлении усовершенствования конструкций настольных механических и электромеханических вычислительных машин и в расширении возможностей так называемых счетно-аналитических машин. Последние получили большое распространение в 1930—1945 гг.

Отличительная особенность счетно-аналитических машин заключается в том, что все исходные числа вместе с указаниями их признаков (рубли, единицы измерений, проценты и т. д.) «записываются» в виде пробитых отверстий на листках плотной бумаги — перфокартах. Устройство для «чтения»

подобной записи состоит из металлических щеточек и контактной пластины или валика, между которыми продвигается перфокарта. Если щеточка попадает на пробивку, то она электрически замыкается с валиком и в соответствующую цепь пойдет ток. Если под щеточкой не оказывается пробивки, то в цепи тока не будет. Другие устройства воспринимают прочитанные электрические импульсы и определяют остальные действия.

Счетно-аналитические машины могут сортировать перфокарты по признакам чисел, выполнять арифметические действия, печатать результаты. Перед каждой очередной обработкой чисел машина настраивается на определенные действия и в дальнейшем эти действия выполняются автоматически.

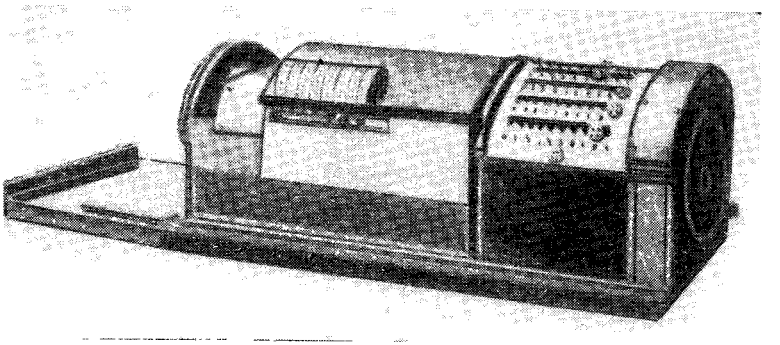


Рис. 3. Общий вид арифмометра Чебышева.

Счетно-аналитические машины не полностью автоматизируют процесс вычислений и доля ручного труда в нем остается значительной. Основными недостатками машин подобного типа являются: малая скорость вычислений и отсутствие способности автоматического перехода с одной части программы на другую или, как говорят, разветвления программы.

Как это часто бывает в истории открытий, в 1937 году Говард Айкен вновь открыл идеи Беббиджа, а в 1938 году Тюринг высказал идею создания универсальной машины. В 1944 году под руководством Айкена была построена релейная цифровая вычислительная машина «Марк I». Предварительно составленная программа вычислений записывалась на бумажную ленту (перфоленту) в виде пробитых отверстий. На другие перфоленты записывались исходные данные. Устройство чтения,

подобное уже описанному для перфокарт, читало записи с перфолент.

По мере движения лент машина последовательно выполняла программу и производила вычисления. Результаты вычислений автоматически пробивались на перфолентах в виде таких же отверстий, как и на перфолентах с исходными данными. Существенным нововведением явилось наличие в машине «памяти» для временного запоминания промежуточных и окончательных результатов. Устройство «памяти» или, как мы будем называть его в дальнейшем, запоминающее устройство, позволяло записывать в него числа, запоминать их и в нужное время выдавать для дальнейших действий. Запоминающее устройство машины могло одновременно хранить до 72 чисел.

«Марк I» могла выполнять различные арифметические и другие действия с большой для того времени скоростью. Так, например, сложение или вычитание двух 23-разрядных десятичных чисел выполнялось за 0,3 сек., умножение — за 6 сек., деление — за 11,4 сек., вычисление $\sin x$ — за 60 сек.

Несмотря на то, что надежность электромеханических*) реле не вызывала на первый взгляд никаких опасений, наладка машины и ее эксплуатация показали обратное. Реле так часто требовали чистки контактов и настройки, что из-за простоев средняя скорость работы машины оказалась очень низкой.

Вскоре была предпринята постройка машины «Марк II», в которой применялись 13 000 реле более надежной конструкции. Однако и эта машина не была достаточно надежной.

При разработке машин «Марк I» и «Марк II» конструкторы столкнулись с трудностью составления релейных схем. Отсутствовала методика составления схем, наиболее экономичных по количеству реле. Одно из удачных решений проблемы было дано американцем Шенноном. Шеннон использовал основные положения алгебры высказываний, или так называемой булевой алгебры, по имени английского математика Буля. Шеннон показал, что булеву алгебру можно применять для анализа и синтеза релейно-контактных схем.

Независимо от Шеннона советским физиком В. И. Шестаковым в 1941 году было предложено применение булевой алгебры к теории релейно-контактных схем.

---

*) Термин «электромеханическое» приводится для того, чтобы подчеркнуть наличие механических подвижных инерционных частей реле. Общепринято — «электромагнитное».

Были предприняты еще несколько попыток создания релейных машин, но каждая новая машина еще больше убеждала конструкторов в малой надежности электромеханических реле.

В 1943 году в Пенсильванском университете была построена первая электронная цифровая вычислительная машина ЭНИАК. Существенно новым в этой машине явилось применение электронных реле. Привычные нам радиолампы в сочетании с другими радиодеталями позволили получить такое реле, которое по скорости переключений и по надежности в работе в сотни и тысячи раз превзошли электромеханические реле. Скажем для сравнения, что наилучший тип электромеханического реле выдерживает около миллиона переключений, после чего приходит в негодность. Хорошее электронное реле делает миллион переключений в одну секунду. Сколько же потребуется сменить электромеханических реле, чтобы выполнить работу электронного реле хотя бы за 500 часов? 1 800 000 штук! А если учесть, что электронные реле сегодняшнего дня имеют срок службы пять и более тысяч часов, то получится куда более внушительное число.

Уникальная по своим размерам машина ЭНИАК содержала около 18 000 электронных ламп и 1500 электромеханических реле. Машина могла в 1 сек. складывать или вычитать 5000 чисел! Управление счетом осуществлялось с помощью программы, набранной вручную на многочисленных коммутационных досках и переключателях. Несоответствие между временем фактического решения задачи и временем подготовки программы вручную было настолько большим, что выигрыш от скорости вычислений почти полностью скрадывался проигрышем во времени на подготовительных операциях.

Решение этой проблемы было найдено Нейманом и опубликовано им в 1947 году. Основная идея заключалась в том, что программу решения задачи следовало вводить в запоминающее устройство машины, после чего машина автоматически выполняла бы всю заданную последовательность действий. Ввод же программы в запоминающее устройство занимает значительно меньшее время. Другая мысль Неймана сводилась к тому, что команды программы, введенные в запоминающее устройство, модифицировались бы самой машиной по мере надобности. Подробно об этом будет сказано в разделе о программировании.

После первых опытов эксплуатации машины ЭНИАК начинается бурное развитие техники электронных цифровых машин.

От машин гигантов конструкторы перешли к созданию небольших машин, более доступных для широкого потребления. Крупнейшие фирмы разных стран, выпускавшие ранее счетно-аналитические машины, телефонное или радиотехническое оборудование, перешли на освоение и выпуск электронных цифровых вычислительных машин. Наиболее известны машины фирм ИБМ, Ферранти, Ремингтон Ренд.

Первая электронная вычислительная машина в Советском Союзе МЭСМ (малая электронная счетная машина) была создана под руководством академика С. А. Лебедева в Институте математики АН УССР. Одной из первых задач, решенных на МЭСМ в 1951 году, была задача передачи электроэнергии на линии Куйбышев — Москва.

В последнее время МЭСМ используется для решения небольших задач и для обучения будущих математиков и инженеров. «Старушка» МЭСМ, как ее теперь любовно называют, дала мощный толчок развитию электронной вычислительной техники в Советском Союзе. Велика заслуга ее создателей, осваивавших новую отрасль техники в тяжелые послевоенные годы.

В 1953 году под руководством академика С. А. Лебедева в Институте точной механики и вычислительной техники АН СССР создается новая универсальная и уникальная по размерам машина БЭСМ (быстродействующая электронная счетная машина). Ряд дополнительных усовершенствований позволил выдвинуть БЭСМ в число наиболее совершенных машин мира. В одну секунду БЭСМ выполняет около 8—10 тысяч арифметических действий и позволяет решать самые разнообразные задачи, такие, как предсказание погоды, перевод текста с одного языка на другой и т. д.

Вскоре наша промышленность освоила выпуск серийной универсальной машины типа «Стрела», созданной под руководством Героя Социалистического Труда Ю. Я. Базилевского. Исключительно разносторонняя и гибкая структура машины, большая скорость вычислений (2—3 тысячи действий в сек.) и колоссальная емкость запоминающих устройств (около 500 000 чисел) позволяют решать на ней практически любые задачи.

Кроме больших универсальных машин типа БЭСМ и «Стрела», был создан целый ряд малых универсальных и специализированных машин: средние машины М-2, М-3, построенные под руководством члена-корреспондента АН СССР И. С. Брука, машины «Урал», «Погода», «Кристалл», СЦМ и др., сконструированные советскими инженерами.

Последнее десятилетие ознаменовалось новыми открытиями, имеющими большое значение для вычислительной техники. Были изобретены запоминающие устройства, основанные на использовании свойств акустических волн, магнитных покрытий, ферромагнитных материалов, электронно-лучевых трубок, магнитострикционного явления, явления поляризации диэлектриков, химического явления фотохромизма и т. д.

Особенное распространение получили запоминающие устройства на магнитной ленте, на магнитном барабане и на сердечниках из ферромагнитных материалов.

Были изобретены быстродействующие магнитные переключатели и реле, использующие свойство проводников при низких температурах резко менять сопротивление под действием магнитного поля (криотроны).

Изобретение в 1948 году полупроводниковых триодов и быстрое их совершенствование, позволяют надеяться, что они скоро займут прочное место в вычислительной технике. К сожалению, широкому использованию полупроводниковых триодов в настоящее время препятствует большая неоднородность характеристик от экземпляра к экземпляру, ограниченные скорости работы и чувствительность к изменению температуры.

В 1956 году были открыты новые усилительные полупроводниковые приборы — спасисторы. Первые испытания спасисторов показали, что они в значительной мере свободны от указанных недостатков.

Однако такие замечательные свойства полупроводниковых триодов, как малые габариты, небольшая потребляемая мощность и большой срок службы заставляют ученых упорно продолжать исследования в этой области. Обратите внимание на рис. 4, где рядом с электронной вакуумной лампой расположилось более десятка полупроводниковых приборов. Уже только разительное отличие в габаритах заставляет с уважением относиться к этим крошечным «животным» на тонких ножках, так быстро и уверенно шагающим во все области техники.

Все больше совершенствовалась методика программирования вычислений, высшей ступенью которой является символическое или автоматическое программирование, требующее минимальной затраты труда математиков.

Необычайно велик круг задач, решаемых электронными цифровыми машинами. Не найдется, пожалуй, такого уголка в науке и технике, где бы вычислительная машина не смогла

оказать человеку существенную помощь. Каждый день приносит сообщения о новых, подчас самых неожиданных ее применениях. Планирование грузооборота на железных дорогах, составление наивыгоднейших маршрутов движения автомобильного транспорта, регулировка движения пешеходов и городского транспорта, бухгалтерские и банковские операции, автоматический пилотаж самолетов, шифровка и расшифровка текста, изучение биологических функций, игра в шахматы, управление

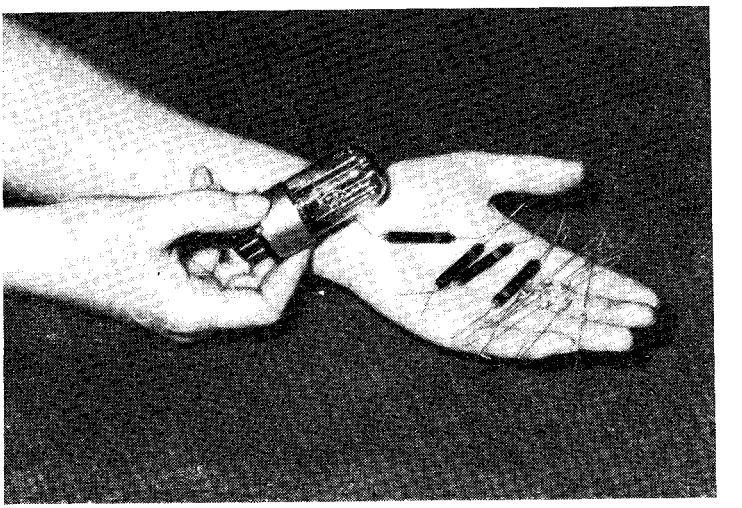


Рис. 4. Полупроводниковые приборы.

сложными производственными процессами — вот очень небольшой перечень применений вычислительных машин.

Рассмотрим подробнее два следующих примера.

Первый пример. Известно, что течение химического процесса может зависеть от многих таких факторов, как температура, давление, состав реагентов и т. д. Если процесс длится многие часы, то есть еще возможность взять пробу промежуточного продукта, и на основе его анализа и последующих математических вычислений установить, каким образом нужно изменить условия реакции с тем, чтобы получить желаемый результат. Но как быть, если реакция длится минутами и времени для подобного анализа нет? В этом случае реакция остается без контроля и нет никакой гарантии

96

качества конечных продуктов. На помощь человеку приходит электронная вычислительная машина и автоматика. В ходе химического процесса автоматические приборы посылают в вычислительную машину данные о температуре, давлении, скорости перемещения реагентов, объеме и т. д. Машина с большой скоростью выполняет все необходимые математические расчеты и посылает в исполнительные механизмы сигналы, согласно которым режимы процесса будут изменены в нужную сторону.

Второй пример. Представьте, что проектируется сложная автоматическая система управления самолетом.

В эту систему входят, как правило, десятки различных автоматических приборов, которые передают вычислительной машине сведения об условиях полета и о состоянии наиболее важных жизненных центров самолета. Как и в предыдущем примере, в систему управления самолетом входят механизмы, которые исполняют приказания вычислительной машины. Сложность взаимодействия всех приборов и машины оказывается настолько большой, что проектировщики часто сами не знают, все ли условия полета учтены, достаточно ли приборов содержит система, допустима ли точность работы проектируемых приборов и т. д. Ответы на все перечисленные вопросы раньше удавалось получить только тогда, когда опытный самолет, оборудованный новой системой, подвергался многочисленным, часто опасным летным испытаниям. При этом в системе нередко обнаруживались самые неожиданные качества, иногда хорошие, а иногда настолько плохие, что вся предварительная работа пропадала даром.

Можно ли избежать последнего, помня о немалых стоимостях экспериментов. Оказывается, можно! Для этого предполагаемые действия приборов и всей системы в целом описываются специальным математическим языком в виде некоторой задачи. Задача составляется в самом общем виде так, что учитываются любые, даже самые невероятные условия полета и состояния системы. Далее задача решается на вычислительной машине и получается так, что машина превращается в модель будущей системы. В спокойных комнатных условиях модель испытывается столько, сколько это нужно для того, чтобы получить уверенность в правильности выбранной системы. Если такая уверенность появится, то можно строить опытный образец. Последний путь создания сложных автоматических систем в десятки раз дешевле первого. К тому же он совершенно безопасен для людей.

2. Основные устройства универсальных и специализированных цифровых машин и их характеристики

Универсальные и специализированные машины имеют много общего по составу устройств. Поэтому мы ограничимся общими замечаниями по поводу тех и других, останавливая внимание только на том, что существенно отличает оба типа машин (общая блок-схема машины изображена на рис. 5).

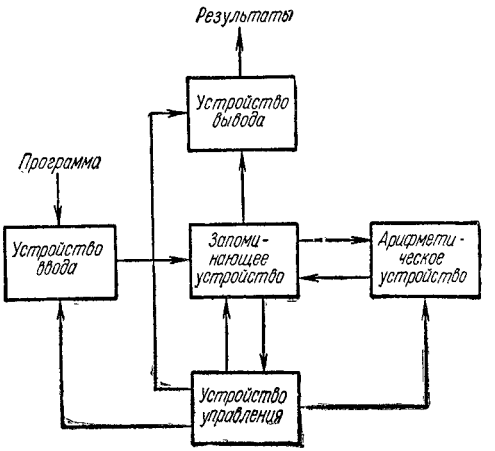


Рис. 5. Общая блок-схема машины.

Из анализа процесса вычислений, осуществляемых человеком, можно заключить следующее. Если любая сложная задача приводится в конечном итоге к последовательности простейших арифметических действий, то универсальная машина должна иметь арифметическое устройство для выполнения этих действий. В функцию **арифметического устройства** универсальной машины входит, как правило, выполнение не только арифметических, но и ряда других, вспомогательных действий (или операций). К числу последних могут относиться: операция нахождения целой части числа, операция нахождения модуля числа, операция сравнения чисел и др.

Основными характеристиками арифметических устройств являются скорость выполнения операций, принцип действия и многообразие видов операций.

Если на арифметическое устройство универсальной машины возлагается требование выполнения большого количества разнообразных операций, то арифметическое устройство специализированной машины может выполнять только те операции, без которых трудно обойтись при решении узкого, заранее известного круга задач.

Вторым, и не менее обязательным, является **запоминающее устройство.** При счете на бумаге нам приходится часто «держать в уме» промежуточные результаты вычислений, если они простые, или записывать их на бумагу, если они сложные. Машина должна уметь делать то же самое и для этого в ее состав входит запоминающее устройство. Запоминающее устройство должно запоминать результаты вычислений и выдавать их для дальнейших действий тогда, когда это предусматривается программой. По аналогии со счетом, выполняемым человеком обычным способом, говорят, что результаты вычислений «записываются» или «считываются» запоминающим устройством. Числовой материал, который нужно запоминать, может быть самым различным как по содержанию, так и по количеству. К нему могут относиться начальные данные задачи, промежуточные и конечные результаты, табличные данные и, наконец, программа вычислений. Программа вычислений содержит перечень инструкций или команд, по которым машина выполняет те или иные операции, и каждая такая команда изображается в машине в виде некоторого числа. Команды-числа записываются и считываются запоминающим устройством так же, как и обыкновенные числа.

Примерно так, как библиотека оценивается по количеству содержащихся в ней книг, запоминающие устройства оцениваются по количеству чисел, которые могут быть записаны или считаны. Эта характеристика запоминающих устройств называется емкостью. Иногда под емкостью подразумевают количество запоминаемых знаков (цифр), а не чисел. Второе, что служит критерием оценки запоминающих устройств, это время, потребное для записи или чтения одного числа, или другими словами, скорость записи или чтения. Наконец, характерным для запоминающего устройства является принцип действия. Последнее в значительной мере определяет степень надежности как данного устройства, так и машины в целом.

В силу особенности своего назначения емкость запоминающих устройств универсальных машин бывает намного больше, чем у специализированных машин. Что касается времени за-

писи и чтения, а также принципа действия, то ни первое, ни второе не является преимущественным для какого-либо одного из рассматриваемых типов машин.

Для того чтобы машина могла решить задачу, ей нужно дать задание в виде программы вычислений. Программа вводится в машину (в запоминающее устройство) с помощью **устройства ввода.** Составляя программу, человек записывает ее вместе с исходными данными сначала на бумаге в виде рукописи. Затем содержание рукописи кодируется или зашифровывается на язык, понятный машине, и записывается на промежуточный носитель. В качестве такого носителя могут быть известные уже нам перфокарты или перфоленты. Устройство ввода «читает» запись с промежуточного носителя и передает ее в запоминающее устройство машины.

И конечно, нужно еще **устройство для вывода** результатов решения задачи. В лучшем случае результаты печатаются устройствами вывода сразу на бумаге в виде десятичных чисел. Но чаще всего результаты выводятся на такие же промежуточные носители, какие использовались при вводе, а затем уже другие устройства, не связанные с вычислительной машиной, печатают их в удобочитаемом виде.

Основной характеристикой устройств ввода и вывода является скорость их работы, которая выражается количеством чисел или цифр, читаемых или печатаемых в одну секунду. К устройствам ввода и вывода универсальных и специализированных машин предъявляется одинаковое требование — большое быстродействие.

Следует заметить, что управляющие вычислительные машины могут не иметь таких устройств ввода и вывода, о которых только что шла речь. Вместо них, очевидно, будут различные датчики, сигнализирующие о состоянии управляемого объекта и исполнительные механизмы, которые будут выполнять приказы машины.

Последним мы назовем **устройство управления.** Само название показывает, что оно должно управлять работой машины или, говоря более точно, управлять работой всех устройств машины так, чтобы выполнялась заданная программой последовательность операций.

II. АРИФМЕТИЧЕСКИЕ ОСНОВЫ ДЕЙСТВИЯ ВЫЧИСЛИТЕЛЬНЫХ МАШИН

1. Изображение чисел с помощью письменных знаков. Системы счисления

Прежде чем познакомиться с работой вычислительной машины, вспомним некоторые положения арифметики целых и дробных чисел. Мы начнем с систем счисления.

Рассмотрим, как изображаются числа при письме с помощью знаков (цифр), ограничиваясь пока случаем целых чисел.

Как известно, наиболее распространенной системой счисления является наша десятичная система. В этой системе любое целое число x может быть однозначно представлено в виде:

$$x = \alpha_{n-1} \cdot 10^{n-1} + \alpha_{n-2} \cdot 10^{n-2} + \cdots + \alpha_1 \cdot 10^1 + \alpha_0 \cdot 10^0 = \\ = \alpha_{n-1}\alpha_{n-2}\cdots\alpha_1\alpha_0,$$

где α_i могут иметь одно из следующих изображений (цифр):

$$0, 1, 2, \cdots, 9.$$

Пример 1. Число «четыре тысячи триста девять» запишется так:

$$4309 = 4 \cdot 10^3 + 3 \cdot 10^2 + 0 \cdot 10^1 + 9 \cdot 10^0.$$

Отметим одну особенность: значение каждой из цифр $1, 2, \ldots, 9$ зависит от ее положения — стоит ли она на месте единиц, десятков, сотен и т. д. Поэтому эта система изображения чисел называется позиционной.

В десятичной системе число «десять» играет особую роль как основание системы. Однако любое другое целое положительное число, большее единицы, можно взять за основание системы счисления.

Так, взяв за основание число «восемь», мы получили бы восьмеричную систему, в которой любое (целое) число представимо однозначно в виде:

$$x = \beta_{n-1} \cdot 8^{n-1} + \beta_{n-2} \cdot 8^{n-2} + \cdots + \beta_1 \cdot 8^1 + \beta_0 \cdot 8^0 = \\ = \beta_{n-1}\beta_{n-2}\cdots\beta_1\beta_0,$$

где β_i могут иметь одно из следующих изображений (цифр):

$$0, 1, 2, 3, 4, 5, 6, 7.$$

Пример 2. Число «семьсот восемьдесят девять» в восьмеричной системе изобразится так:

$$789 = 1 \cdot 8^3 + 4 \cdot 8^2 + 2 \cdot 8^1 + 5 \cdot 8^0,$$

т. е.

$$789 = (1425)_8.$$

Такую запись, где цифра внизу скобок в правой части обозначает основание системы, мы будем применять и в дальнейшем. Основание же «десять» здесь и далее писать не будем.

Существует простое правило перехода от основания «десять» к любому основанию d: надо произвести последовательные деления данного числа x на d; полученные остатки, начиная с последнего, и будут «цифрами» при записи числа x в системе с основанием d. *)

Так,

789	8		
5	98	8	
	2	12	8
		4	1

остатки 1, 4, 2 и 5 и есть цифры, изображающие число 789 в восьмеричной системе.

В современной машинной математике особое место заняла система записи чисел с основанием «два». Это связано с тем, что многие элементы быстродействующих машин, как например, электронное реле, могут иметь два устойчивых состояния: «включено» и «выключено». Если теперь условимся одно из состояний такого элемента, например «включено», обозначить единицей, а другое состояние нулем, то удобство изображения чисел в машине в двоичной системе станет понятным.

*) Доказательство в виде упражнения предоставляем читателю.

В двоичной системе применяются только две цифры: 0 и 1. Таким образом, любое (целое) число представляется в виде:

$$x = \alpha_{k-1} \cdot 2^{k-1} + \alpha_{k-2} \cdot 2^{k-2} + \cdots + \alpha_1 \cdot 2 + \alpha_0 \cdot 2^0 = \\ = \alpha_{k-1}\, \alpha_{k-2} \cdots \alpha_1\, \alpha_0,$$

где

$$\alpha_i = 0 \text{ или } 1 \qquad (i = 0, 1, 2, \ldots, k-1).$$

Пользуясь приведенным выше правилом, читатель без труда сможет перевести запись любого (целого) числа из десятичной системы в двоичную.

Приведем все же таблицу нескольких первых чисел натурального ряда (вместе с числом 0), записанных в десятичной

Таблица 1

Десятичная	Двоичная	Десятичная	Двоичная	Десятичная	Двоичная	Десятичная	Двоичная
0	0	6	110	12	1100	18	10010
1	1	7	111	13	1101	19	10011
2	10	8	1000	14	1110	20	10100
3	11	9	1001	15	1111	21	10101
4	100	10	1010	16	10000	22	10110
5	101	11	1011	17	10001	23	10111

и двоичной системах (табл. 1). Одно из неудобств практического применения двоичной системы видно из рассмотрения этой таблицы: для изображения даже небольших чисел нужны длинные выражения. Но при работе на быстродействующих машинах эта система, как указывалось, имеет очевидные технические преимущества. Однако не только это делает удобным применение двоичной системы в быстродействующих машинах: такие арифметические действия, как сложение и умножение, особенно просто выполняются именно в двоичной системе.

Сложение происходит по следующим правилам:

$$0 + 0 = 0, \quad 1 + 0 = 0 + 1 = 1, \quad 1 + 1 = 10$$

(в последнем случае должен быть осуществлен перенос единицы из второго разряда суммы в следующий старший разряд).

Пример 3. Сложения двух чисел, записанных в двоичной системе:

$$\begin{array}{r} 101\,001 \\ +\ 110\,101 \\ \hline 1\,011\,110 \end{array}$$

или то же в десятичной:

$$41 + 53 = 94, \quad 94 = (1\,011\,110)_2.$$

Правила вычитания также достаточно просты:

$$0 - 0 = 0, \quad 1 - 0 = 1, \quad 1 - 1 = 0, \quad 10 - 1 = 1,$$

где 10 — основание двоичной системы.

Умножение происходит по таким правилам:

$$0 \cdot 0 = 0, \quad 0 \cdot 1 = 1 \cdot 0 = 0, \quad 1 \cdot 1 = 1.$$

Как видим, по сравнению с громоздкой десятичной таблицей умножения двоичная достаточно проста.

Пример 4. Умножение двух чисел, записанных в двоичной системе:

$$\begin{array}{r} \times\ 1011 \\ \quad\ 101 \\ \hline \quad\ 1011 \\ +\ 1011 \\ \hline \quad 110111 \end{array}$$

или в десятичной

$$11 \times 5 = 55, \quad 55 = (110\,111)_2.$$

Из этого примера легко заметить, что действие умножения в двоичной системе сводится к последовательному сдвигу числа на несколько разрядов влево и сложению.

Между восьмеричной и двоичной системами существует простая и важная связь, которая позволяет переводить запись числа из восьмеричной в двоичную и обратно простым способом. Разберем это на примере. Пусть задано число в двоичной системе

$$101\ \ 111\ \ 001.$$

Разобьем это число на группы (справа налево) так, чтобы в каждой группе было по три разряда:

$$101\ \ 111\ \ 001.$$

Каждая группа представляет одну из цифр в восьмеричной системе. В нашем случае это будет:

$$5\ \ 7\ \ 1,$$

таким образом,

$$(101\,111\,001)_2 = (571)_8.$$

Наоборот, пусть, например, число 126, записанное в восьмеричной системе, надо перевести в двоичную. Для этого каждую цифру этого числа представляем в двоичном виде:

$$001\ 010\ 110.$$

Таким образом,

$$(126)_8 = (1\,010\,110)_2$$

(нули перед старшим разрядом опускаем).

Теперь ясно, что для перевода чисел из десятичной системы в двоичную удобнее сначала перевести десятичную запись в восьмеричную, которая уже просто переводится в двоичную.

При работе на быстродействующих машинах часто используется так называемая двоично-десятичная система. В этой системе каждая из десятичных цифр 0, 1, 2, ..., 9 изображается в двоичной системе, для чего отводится четыре разряда на каждую десятичную цифру. Например, число 2795 в двоично-десятичной записи имеет вид:

$$\underbrace{0010}_{2}\ \underbrace{0111}_{7}\ \underbrace{1001}_{9}\ \underbrace{0101}_{5}.$$

Таким образом,

$$2795 = (0010\,0111\,1001\,0101)_{2;\ 10}.$$

Все сказанное о системах счисления касалось арифметики целых чисел. Легко распространить это и на случай дробных чисел, если учесть, что, например, дробь

$$x = \alpha_3\ \alpha_2\ \alpha_1\ \alpha_0,\ \alpha_{-1}\ \alpha_{-2}\ \ldots\ \alpha_{-n}$$

может быть изображена так:

$$x = \alpha_3 \cdot 10^3 + \alpha_2 \cdot 10^2 + \alpha_1 \cdot 10^1 + \alpha_0 \cdot 10^0 + \\ + \alpha_{-1} \cdot 10^{-1} + \ldots + \alpha_{-n} \cdot 10^{-n}.$$

Пример 5.

$$31{,}829 = 3 \cdot 10^1 + 1 \cdot 10^0 + 8 \cdot 10^{-1} + 2 \cdot 10^{-2} + 9 \cdot 10^{-3}.$$

Таким образом, запятая отделяет отрицательные показатели основания системы от положительных и нулевого. Заметим, что это справедливо и для системы счисления с произвольным основанием d.

Как изобразить дробное число в двоичной системе?

Сперва рассмотрим случай, когда знаменатель есть целая степень двойки.

Пример 6. Пусть надо записать в двоичной системе дробь

$$5\frac{7}{8}=5+\frac{7}{8}.$$

Имеем:

$$5=(101)_2;\ \frac{7}{8}=\frac{1\cdot 2^2+1\cdot 2^1+1\cdot 2^0}{2^3}=1\cdot 2^{-1}+1\cdot 2^{-2}+1\cdot 2^{-3}.$$

То есть

$$5\frac{7}{8}=(101{,}111)_2.$$

Теперь пусть знаменатель не есть целая степень двойки. В этом случае можно поступить так, как поступают при переводе из обыкновенной дроби в десятичную, т. е. делением числителя на знаменатель. Разумеется, при этом могут получиться бесконечные дроби, как и в десятичной системе.

Пример 7. Пусть дробь $\frac{3}{7}$ надо записать в двоичной системе. Запишем числитель и знаменатель в двоичной системе, будем иметь:

$$\frac{3}{7}=\left(\frac{11}{111}\right)_2.$$

Произведем деление числителя на знаменатель, используя правила умножения и вычитания:

$$\begin{array}{r|l}
11 & 111 \\ \cline{2-2}
110 & 0{,}0110110\ldots \\
-\ 1100 & \\
111 & \\ \cline{1-1}
-\ 1010 & \\
111 & \\ \cline{1-1}
-\ 1100 & \\
111 & \\ \cline{1-1}
-\ 1010 & \\
111 & \\ \cline{1-1}
\ldots &
\end{array}$$

Проконтролируем результат (который, как видно, есть бесконечная периодическая дробь) путем перевода его снова в

десятичную систему:

$$(0,\ 011\ 011)_2 = \frac{27}{64}$$

$\frac{27}{64}$ отличается от $\frac{3}{7}$ меньше чем на 10^{-2}, что объясняется сделанным округлением.

После всего сказанного о системах счисления становится ясной запись любого числа в системе с произвольным основанием d в компактной форме:

$$x = \pm \sum_{i=1}^{i=r} \alpha_i \cdot d^{\tau - i}, \qquad (1)$$

где τ — заданное фиксированное число, определяющее положение запятой, r — число разрядов *).

В этом разделе были рассмотрены случаи, когда $d = 2, 8, 10$ (десять).

Из примера 7 видно, что перевод из десятичной системы в двоичную довольно трудоемок и неудобен. В цифровых автоматах этот перевод производится автоматически и с большой скоростью. Предварительно числа должны быть переведены в двоично-десятичную систему. Последнее осуществляется вне машины на специальном устройстве.

Подведем кратко итог нашего знакомства с различными системами. Число, записанное в десятичной системе, может быть записано в восьмеричной, двоичной и двоично-десятичной системе. Перевод из восьмеричной в двоичную, как мы убедились, прост. Проста по своему принципу и запись числа в двоично-десятичном виде.

Существуют и другие системы счисления, применяемые в электронных вычислительных машинах. Так, в автоматической цифровой машине М-2 Лаборатории управляющих машин и систем Академии наук СССР применяется шестнадцатеричная система. Ведутся исследования по созданию элементов, могущих занимать три устойчивых состояния, что позволит применить троичную систему. Однако подробно на этом мы останавливаться не можем.

---

*) Знак $\sum_{i=1}^{i=r}$ часто употребляется в математике; он обозначает сумму ряда слагаемых, каждое из которых записано как одно и то же выражение, но с разными значками (индексами) при этом выражении.

Упражнение 1

1) Запишите число 3071 вначале в восьмеричной, а затем в двоичной системе.

2) Запишите Ваш год рождения в двоичной системе.

3) Запишите число 574 в троичной, и затем в девятеричной системе.

4) Запишите число $3\frac{5}{32}$ в двоичной системе.

5) Запишите число 5, 7 в двоичной системе, ограничиваясь 8 знаками после запятой. Затем проконтролируйте результат, переведя его в десятичную систему.

6) Переведите бесконечную периодическую двоичную дробь 1101,(101) в бесконечную периодическую десятичную дробь.

2. Нормальная форма представления чисел. Плавающая и фиксированная запятая

В предыдущем разделе мы познакомились с различными системами счисления, применяемыми в цифровых машинах. Однако сами числа, в какой бы системе они ни были записаны, могут быть по-разному представлены в самой машине. В самом деле, при решении задачи как промежуточные, так и окончательные результаты могут быть весьма различны по своей величине. При этом возникает вопрос: как узнавать положение запятой, отделяющей целую часть числа от дробной? Возможны два подхода к разрешению этой задачи: 1) либо отвести запятой постоянное место, 2) либо по мере надобности менять ее положение.

При первом способе представления чисел это соответствует тому, что в равенстве (1) число τ задано наперед и фиксировано. Существуют машины, для которых $\tau = 0, 1, 2, 3$. Последнее и означает, что запятая занимает постоянное место. Все числа, получающиеся в процессе счета в таких машинах (они называются машинами с фиксированной запятой), должны быть всегда ограничены, то есть должны быть меньше некоторого постоянного для данной машины числа. Если получаются числа, большие чем это число, то машина «теряет» старшие разряды таких чисел, происходит, как говорят, переполнение разрядной сетки. Поэтому в машинах с фиксированной запятой нужно выбирать особые схемы решения задач, дабы промежуточные результаты не дали переполнения раз-

рядной сетки. Это является некоторым неудобством таких машин.

При втором способе представления, когда запятая меняет свое место, τ меняется от числа к числу. Машины, где применяется этот способ представления чисел, называются машинами с плавающей запятой.

Все числа в таких машинах представляются в виде:

$$x = M_x \cdot d^{\pm p}, \tag{2}$$

где по-прежнему d — основание системы счисления.

В равенстве (2) M_x называется мантиссой числа x и для нее должны соблюдаться неравенства:

$$d^{-1} \leqslant M_x < 1{,}0; \tag{3}$$

p — целое число, называемое порядком (с соответствующим знаком).

Форма представления числа (2) называется нормальной, а при соблюдении неравенств (3) число x называется нормализованным. Разъясним запись (2) на примерах. В десятичной системе, когда $d = 10$, нужно, чтобы $\frac{1}{10} \leqslant M_x < 1{,}0$, т. е. мантисса должна быть правильной дробью, имеющей после запятой одну из цифр 1, 2, ..., 9.

Пример 1. Число $x = 25{,}3798$ в нормальной форме и нормализованное будет $x = 0{,}253798 \cdot 10^{+2}$. Здесь $p = +2$ и $M_x = 0{,}253798$.

Пример 2. $x = 0{,}0031452$. В нормальной форме $x = 0{,}31452 \cdot 10^{-2}$. Здесь $p = -2$ и $M_x = 0{,}31452$.

В двоичной системе, когда $d = 2$, нужно, чтобы

$$\frac{1}{2} \leqslant M_x < 1{,}0.$$

Пример 3. $x = 5\frac{7}{8}$ в нормальной форме в двоичной системе будет $x = 0{,}101111 \cdot 2^{11}$. Здесь $p = 11$ (три в десятичной системе) и $M_x = 0{,}101111$ (т. е. M_x — нормализованное число).

Таким образом, чтобы задать число в машине с плавающей запятой, необходимо задать p и M_x. Если, например, $x = \pm \alpha_1 \ \alpha_2 \ \ldots \ \alpha_k \cdot 10^{\pm \tau_1 \tau_2 \cdots \tau_m}$, где 10 есть основание двоичной системы, а α_i и τ_i есть либо 0, либо 1, то такое число можно изобразить двумя группами цифр, из которых

первая группа будет изображать мантиссу с ее знаком, а вторая группа — порядок числа x со своим знаком. Такое распределение разрядов между этими двумя группами в машине с плавающей запятой называется *разрядной сеткой*.

Таким образом, любое число на разрядной сетке изобразится так:

α_0	$\alpha_1\ \alpha_2\ \ldots\ldots\ \alpha_k$	τ_0	$\tau_1\ \tau_2\ \ldots\ \tau_m$

где α_0 — означает знак числа x ($\alpha_0 = 1$, если $x < 0$ и $\alpha_0 = 0$, если $x > 0$);

τ — знак порядка ($\tau_0 = 1$, если порядок отрицательный и $\tau_0 = 0$, если он положителен).

Если число x нормализовано, т. е. M_x лежит в пределах, указанных неравенствами (3), то $\alpha_1 \neq 0$. Если же M_x лежит вне этих пределов, то x называется ненормализованным. Нормализация чисел в машинах может производиться при вводе чисел в машину, при выполнении арифметических действий, а также может осуществляться с помощью особой операции, которая называется сдвиг. Сущность этой операции заключается в том, что все разряды мантиссы сдвигаются так, чтобы абсолютная величина ее попала внутрь интервала (3). При этом если сдвиг происходит вправо, то порядок увеличивается на столько единиц, на сколько разрядов произошел сдвиг мантиссы. При сдвиге влево величина порядка уменьшается на соответствующее число единиц.

Пример 4. Пусть число записано в двоичной системе, в нормальной форме, но не нормализовано:

$$x = 0{,}00101 \cdot 2^{-110}.$$

Нормализуем его путем сдвига мантиссы влево на два разряда, порядок при этом уменьшится на две единицы. После нормализации будем иметь:

$$x = 0{,}101 \cdot 2^{-1000}.$$

Каждая из машин с фиксированной и плавающей запятой имеет свои преимущества и недостатки. Основной недостаток машин с фиксированной запятой, как уже отмечалось, это необходимость выбора специальных схем счета задач. Но зато эти машины проще в управлении и дешевле в изготовлении. Машины с плавающей запятой свободны от этого недостатка,

но сложнее в эксплуатации и дороже, так как требуют дополнительного оборудования для порядков чисел.

У нас в Союзе машинами с плавающей запятой являются уже упомянутая нами БЭСМ и «Стрела». Машина М-2 способна работать в двух режимах: с плавающей и с фиксированной запятой.

Дальнейшее изложение касается в основном машин с плавающей запятой, причем за основу принята машина типа «Стрела». Это не значит, конечно, что мы здесь расскажем только о «Стреле». Ведь ЭЦВМ*) разных типов имеют много общих черт, и, зная одну машину, легко разобраться в принципах работы другой.

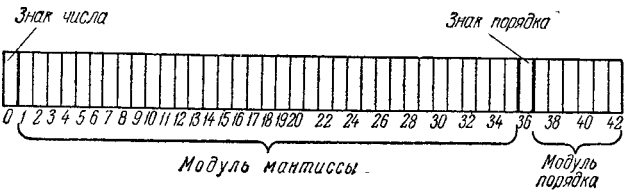


Рис. 6. Разрядная сетка «Стрелы».

В машинах «Стрела» для изображения чисел отведено 43 разряда (с 0-го по 42-й, нумерация слева направо). Нулевой разряд отведен для знака числа, а 36-й разряд для знака порядка, причем знак «плюс» изображается нулем, а знак «минус» единицей. Таким образом, мантисса числа, записанная в двоичной форме, занимает 35 разрядов (с 1-го по 35-й включительно). Разряды с 37-го по 42-й отведены для порядков чисел, изображаемых также в двоичной форме. В десятичной системе это соответствует диапазону чисел примерно от 10^{-19} до 10^{+19}, достаточно большому для любых вычислений**).

Это размещение числа в разрядной сетке изображено схематично на рис. 6.

---

*) Так здесь и в дальнейшем мы обозначаем электронные цифровые вычислительные машины.

**) Действительно, наибольшее по абсолютной величине число на такой разрядной сетке будет изображаться единицами во всех разрядах мантиссы и порядка. При этом мантисса будет $1 - 2^{-35}$, а порядок 63. Следовательно, все число будет $(1 - 2^{-35}) \cdot 2^{63} \approx 2^{63}$. Это число несколько меньше, чем 10^{19}.

3. Об изображении отрицательных чисел

Чтобы произвести алгебраическое сложение двух чисел A и B

$$(\pm A) + (\pm B),$$

нужно предусмотреть в арифметическом устройстве машины возможность выполнения операций сложения и вычитания. Оказывается, что, модифицируя по особым правилам запись отрицательного числа, можно операцию вычитания свести к сложению. В этом разделе мы сформулируем правила, по которым производится эта модификация, а в следующем разделе расскажем о том, как применяются эти правила при выполнении арифметических операций.

В машинах для изображения отрицательных чисел применяются три способа их записи: прямой, обратный и дополнительный коды. Нам понадобятся в дальнейшем прямой и обратный коды. Их мы здесь и рассмотрим.

Читатель неоднократно имел дело с обратным кодом, вероятно, не подозревая этого, когда производил вычисления с помощью десятичных логарифмов. Действительно, если рассматривать числа, являющиеся правильными положительными дробями, то логарифмы таких чисел отрицательны. Однако при вычислениях, связанных с использованием таблиц десятичных логарифмов, всегда логарифмы чисел записываются так, чтобы мантиссы были бы положительны. Например, $\log 0{,}3 = -0{,}52288$, но пишут так:

$$\log 0{,}3 = \bar{1}{,}47712 = -1 + 0{,}47712 = -0{,}52288$$

(положительную мантиссу 0,47712 находят прямо из таблиц). Запись $\bar{1}\,,47712$ можно рассматривать как условную запись числа $-0{,}52288$, т. е. код этого числа. Если отбросить черту над характеристикой, то 1 в такой записи означает знак числа (минус), а положительная мантисса 0,47712 есть дополнение числа 0,52288 до единицы, т. е. $0{,}47712 = 1 - 0{,}52288$.

В двоичной системе такое дополнение, как легко проверить, можно получить очень просто: надо единицы заменить нулями, а нули единицами и прибавить единицу в младший разряд.

В машинах, где применяется обратный код, такое прибавление не происходит, а осуществляется только указанная выше замена нулей единицами и наоборот. Это равносильно тому, что дополнение получается вычитанием не из единицы, а из $1 - 10^{-n}$, где n — число разрядов в разрядной сетке,

а 10 — основание двоичной системы. Это и есть запись отрицательного числа x ($|x|<1$) в обратном коде.

После этих наводящих соображений перейдем к формулировке правил образования прямого и обратного кодов. Число x, записанное в этих кодах, будет обозначать соответственно

$$(x)_{\text{пр}},\ (x)_{\text{обр}}.$$

Здесь мы будем рассматривать числа, записанные в двоичной системе счисления и являющиеся правильными дробями, т. е. числа вида:

$$x = \pm 0,\ \alpha_1\alpha_2 \ldots \alpha_k, \tag{4}$$

где α_i есть либо 0, либо 1.

Для положительных чисел изображение в этих двух кодах совпадает с записью (4), т. е. если $x>0$, то

$$(x)_{\text{пр}} = (x)_{\text{обр}} = x = 0,\ \alpha_1\alpha_2 \ldots \alpha_k.$$

Перейдем к отрицательным числам. Начнем с прямого кода.

Правило. Чтобы написать в прямом коде отрицательное число, нужно дробную часть его оставить без изменения, а вместо знака «минус» записать единицу и поставить ее на место нуля целых.

Таким образом, если $x<0$, то $(x)_{\text{пр}} = 1,\ \alpha_1\alpha_2 \ldots \alpha_k$.

Рассмотрим обратный код. Предварительно определим операцию логического отрицания, применяемую к числам 0 и 1: $\overline{0}=1$, $\overline{1}=0$ (здесь черта над числом означает логическое отрицание). Теперь сформулируем следующее.

Правило. Чтобы написать в обратном коде отрицательное число, нужно вместо знака минус написать единицу, поставив ее на место нуля целых, а к дробной части применить поразрядное логическое отрицание.

Таким образом, если $x<0$, то

$$(x)_{\text{обр}} = 1,\ \overline{\alpha}_1\overline{\alpha}_2 \ldots \overline{\alpha}_k,$$

где $\overline{\alpha}_i = \begin{cases} 0, & \text{если } \alpha_i = 1, \\ 1, & \text{если } \alpha_i = 0. \end{cases}$

Пример 1. $x = -0{,}1101$, $(x)_{\text{обр}} = 1,\overline{1}\,\overline{1}\,\overline{0}\,\overline{1} = 1{,}0010$.

Перейдем теперь к изучению арифметических операций.

4. Арифметические действия, совершаемые машиной

В большинстве ЭЦВМ предусмотрена возможность выполнения всех четырех арифметических действий, так как решение любой сложной математической задачи на практике

сводится к некоторой последовательности этих четырех действий.

В этом разделе мы разберем, как происходят действия сложения, вычитания и умножения, опуская, ради простоты, операцию деления.

Сложение

В случае сложения одно слагаемое или оба одновременно могут быть числами не нормализованными, мантиссы таких чисел лежат вне пределов, указанных неравенствами (3). Результат сложения во всех случаях получается нормализованным. Этим, в частности, можно воспользоваться для нормализации какого-либо числа, стоит лишь сложить его с нулем.

Итак, пусть одно слагаемое есть $x = M_x \cdot 2^{p_x}$, а другое $y = M_y \cdot 2^{p_y}$, где M_x и M_y — мантиссы, а p_x и p_y — порядки чисел x и y. Их сумма:

$$x + y = M_x \cdot 2^{p_x} + M_y \cdot 2^{p_y}.$$

Пусть $p_x = p_y = p$; тогда

$$x + y = (M_x + M_y) \cdot 2^p,$$

и действие сложения свелось к сложению мантисс. Если мантисса суммы $M_x + M_y$ не есть нормализованное число, то следует нормализовать ее путем сдвига. При этом соответственно изменится и величина порядка результата.

Пусть теперь $p_x \neq p_y$, например $p_x > p_y$. В этом случае прежде всего следует выравнять порядки, либо меньший увеличить до большего, либо наоборот. Условимся в дальнейшем меньший порядок увеличивать до бóльшего.

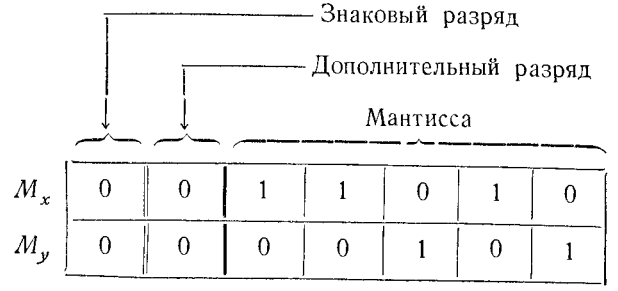
После определения бóльшего порядка мантисса числа с меньшим порядком сдвигается вправо на число разрядов, равное $|p_x - p_y|$. Затем происходит сложение мантисс и нормализация мантиссы результата, если это нужно.

Сказанное проиллюстрируем примерами.

Пример 1. Пусть $x = 0{,}1101 \cdot 2^{101}$ и $y = 0{,}1010 \cdot 2^{11}$. Найдем сумму $x + y$. Порядки слагаемых не равны, поэтому первый шаг будет заключаться в выравнивании меньшего порядка $p_y = 11$ до бóльшего $p_x = 101$, для чего мантиссу M_y следует сдвинуть на $p_x - p_y = 10$ (два) разряда вправо. После сдвига $p_x = p_y = 101$ и $x = 0{,}1101 \cdot 2^{101}$, а $y = 0{,}00101 \cdot 2^{101}$.

Прежде чем приступить ко второму шагу — сложению мантисс M_x и M_y, введем небольшое усложнение в знакомую

нам разрядную сетку, а именно: между разрядом, отведенным для знака числа и старшим разрядом мантиссы предусмотрим еще один дополнительный разряд, назначение которого будет ясно из дальнейшего (в нем сначала записан нуль). Запишем мантиссы M_x и M_y наших чисел на таких сетках. Напомним, что знак «плюс» изображается нулем, знак «минус» единицей.



Теперь сложим эти две двоичные записи, оперируя со знаковым и дополнительным разрядами как с обычными цифрами двоичного числа. Получим:

0	0	1	1	1	1	1

Это и есть мантисса суммы. Она, как видно, есть нормализованное число. Таким образом:

$$x+y=0{,}11\ 111\cdot 2^{101}.$$

Пример 2. $x=0{,}11010\cdot 2^0$, $y=0{,}10011\cdot 2^0$.

M_x	0	0	1	1	0	1	0
M_y	0	0	1	0	0	1	1
	0	1	0	1	1	0	1

Здесь мантисса суммы не нормализована. (Единицу старшего разряда суммы, не поместившуюся в разрядах мантиссы, мы поместили в дополнительный разряд.) Для нормализации

следует произвести сдвиг так, чтобы единица из дополнительного разряда попала в старший разряд мантиссы. Так как сдвиг происходит здесь на один разряд вправо, то, увеличив порядок на единицу, окончательно получим:

$$x+y=0{,}101\ 101\cdot 2^{+1}.$$

Пример 3. Сложим ненормализованные числа $x=0{,}00101\cdot 2^{0}$, $y=0{,}00011\cdot 2^{0}$.

M_x	0	0	0	0	1	0	1
M_y	0	0	0	0	0	1	1
	0	0	0	1	0	0	0

Чтобы нормализовать мантиссу результата в этом случае, следует сдвиг произвести влево, пока единица мантиссы не попадет в старший разряд. При сдвиге влево новые появляющиеся разряды мантиссы записываются как нули.

Окончательно:

$$x+y=0{,}100\ 00\cdot 2^{-1}.$$

Пример 4. $x=0{,}1101\cdot 2^{0}$, $y=0{,}1110\cdot 2^{0}$.

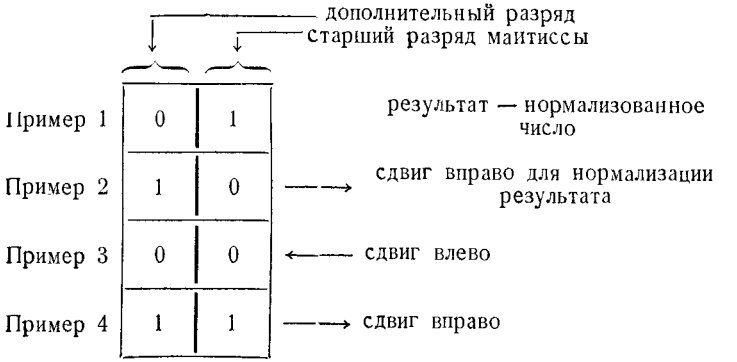
M_x	0	0	1	1	0	1	0
M_y	0	0	1	1	1	0	0
	0	1	1	0	1	1	0

После сдвига мантиссы вправо на один разряд получим нормализованный результат:

$$x+y=0{,}11011\cdot 2^{+1}.$$

Эти четыре примера исчерпывают все возможные комбинации нулей и единиц в дополнительном и старшем разрядах мантиссы, которые могут получаться при сложении двух поло-

жительных чисел. Действительно, наши четыре примера образуют следующую схему:



Сформулируем теперь основные правила сложения:

1) Из порядка первого слагаемого p_x вычитается порядок второго p_y и анализируется знак разности $p_x - p_y$, чтобы найти меньшее число.

2) Мантисса меньшего числа сдвигается вправо на число разрядов, равное $|p_x - p_y|$.

3) Происходит сложение мантисс.

4) Если нужно, результат нормализуется.

Читателю рекомендуется самостоятельно проверить еще раз эти правила на следующих примерах.

Упражнение II

Сложить числа:

1) $x = 0{,}1010 \cdot 2^{+110}$, $y = 0{,}1100 \cdot 2^{+11}$;
2) $x = 0{,}1100 \cdot 2^{0}$, $y = 0{,}0010 \cdot 2^{0}$;
3) $x = 0{,}0011 \cdot 2^{0}$, $y = 0{,}0100 \cdot 2^{0}$;
4) $x = 0{,}1101 \cdot 2^{0}$, $y = 0{,}1110 \cdot 2^{0}$.

Вычитание

Вычитание является сложением двух чисел разных знаков. Выше мы рассказали о том, что для изображения отрицательных чисел применяется особый способ их записи — кодирование, благодаря чему операция вычитания может быть сведена к сложению. Мы проследим здесь на конкретных примерах, как это происходит.

Для простоты будем рассматривать числа с одинаковыми порядками, так как в противном случае при вычитании, как и при сложении, первый шаг заключается в выравнивании порядков — операции нам уже знакомой.

Так же, как и при сложении, одно или два числа, уменьшаемое и вычитаемое, могут быть ненормализованными числами. Результат получается всегда нормализованным.

Пример 1. Пусть $x = 0{,}10110 \cdot 2^0$ и $y = -0{,}11011 \cdot 2^0$. Сложим эти два числа, т. е. произведем действие вычитания. Имеем:

M_x	0	0	1	0	1	1	0
M_y	1	0	1	1	0	1	1

Отрицательное число y переписываем в обратном коде, применяя поразрядное логическое отрицание ко всем разрядам, кроме знакового. После этого производим известное нам сложение мантисс:

M_x	0	0	1	0	1	1	0
$(M_y)_{\text{обр}}$	1	1	0	0	1	0	0
	1	1	1	1	0	1	0

Так как в знаковом разряде мантиссы результата стоит единица, то, снова применяя поразрядное логическое отрицание, получим:

1	0	0	0	1	0	1

Это и есть мантисса результата. Ее следует только нормализовать, что достигается сдвигом влево на два разряда. Окончательно:

$$x + y = 0{,}10110 \cdot 2^0 - 0{,}11011 \cdot 2^0 = -0{,}10100 \cdot 2^{-10}.$$

При вычитании может быть случай, когда единица переноса из дополнительного разряда должна попасть в знаковый, где

стоит уже единица (если наши числа разных знаков!) и, следовательно, после сложения с этой знаковой единицей новая единица переноса выйдет за разрядную сетку *). Тогда эта единица прибавляется к младшему разряду мантиссы (так называемый циклический перенос).

Проиллюстрируем это на примере.

Пример 2. $x = 0{,}10110 \cdot 2^0$, $y = -0{,}01011 \cdot 2^0$.

По обычным правилам

M_x	0	0	1	0	1	1	0
$(M_y)_{\text{обр}}$	1	1	1	0	1	0	0
	0	0	0	1	0	1	0
1							$+1$

В знаковом разряде результата — нуль. В этом случае поразрядное логическое отрицание не происходит, и после нормализации получаем окончательный ответ:

$$x + y = 0{,}10110 \cdot 2^0 - 0{,}01011 \cdot 2^0 = 0{,}10110 \cdot 2^{-1}.$$

Сформулируем теперь основные правила вычитания:

1) Из порядка p_x одного числа вычитается порядок p_y другого числа и анализируется знак числа $p_x - p_y$, чтобы найти меньшее.

2) Мантисса меньшего числа сдвигается вправо на число разрядов, равное $|p_x - p_y|$.

3) Отрицательные числа переписываются в обратном коде, для чего применяется поразрядное логическое отрицание ко всем разрядам, кроме знакового.

4) Складываем мантиссы. В случае выхода единицы за разрядную сетку последняя прибавляется к младшему разряду мантиссы.

5а) Если в знаковом разряде результата стоит единица, то снова применяем поразрядное логическое отрицание и нормализуем результат, если нужно.

5б) Если в знаковом разряде результата стоит нуль, то сразу нормализуем результат, если это нужно.

*) Легко сообразить, что при сложении двух положительных чисел такого случая быть не может.

Упражнение III

Проверьте правила вычитания на следующих примерах:
1) $x = 0{,}1101 \cdot 2^{+1}$; $y = -0{,}1011 \cdot 2^{+10}$;
2) $x = -0{,}1001 \cdot 2^{+111}$, $y = 0{,}0110 \cdot 2^{+11}$.

Умножение

При этом действии в машинах с плавающей запятой оба сомножителя должны быть нормализованными числами. В противном случае ненормализованное число воспринимается машиной как нуль.

Если

$$x = \pm M_x \cdot 2^{p_x}$$

и

$$y = \pm M_y \cdot 2^{p_y},$$

то

$$x \cdot y = \pm M_x \cdot M_y \cdot 2^{p_x + p_y},$$

где $p_x + p_y$ — алгебраическая сумма порядков.

Как видно, действие умножения можно проанализировать, рассматривая только умножение мантисс, так как алгебраическое суммирование порядков p_x и p_y ничем не отличается от известных уже нам правил сложения и вычитания мантисс.

Начнем с определения знака произведения. Он определяется по следующей схеме:

$0 + 0 = 0$ (т. е. произведение двух положительных чисел положительно),

$0 + 1 = 1$, $1 + 0 = 1$ (произведение двух чисел разных знаков отрицательно),

$1 + 1 = 0$ (произведение двух отрицательных чисел положительно; перенос единицы в старший разряд здесь не происходит).

После того как определен знак произведения, происходит алгебраическое сложение порядков. Последний этап — перемножение мантисс и нормализация результата, если это нужно.

Пример 1. $x = +0{,}1101 \cdot 2^{+11}$, $y = -0{,}1011 \cdot 2^{-1}$.

Согласно сказанному знак произведения будет:

$$0 + 1 = 1 \quad \text{(минус)}.$$

Складываем порядки: $+11 - 1 = +10$.

Перемножаем модули мантисс по уже знакомым нам правилам:

$$\begin{array}{r} \times\ 0{,}1101 \\ 0{,}1011 \\ \hline 1101 \\ +\ 1101 \\ 0000 \\ 1101 \\ \hline 0{,}10001111 \end{array}.$$

(Положение запятой определяем обычным образом, как и в десятичном случае.)

Так как мантисса результата не требует нормализации, то окончательно будем иметь:

$$x \cdot y = 0{,}10001111 \cdot 2^{+10}.$$

Упражнение IV

Перемножьте числа, применяя последовательность операций указанных в этом разделе:

1) $x = -11{,}0101 \cdot 2^{+10}$, $y = 0{,}01101 \cdot 2^{-1}$;
2) $x = 1{,}00011 \cdot 2^{-11}$, $y = 10{,}11001 \cdot 2^{+10}$.

III. ЭЛЕМЕНТЫ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ

1. Логические элементы

Так же, как строители из нескольких типов кирпичей или других строительных материалов возводят здания самой различной архитектуры, в цифровой вычислительной технике используются всего несколько типов элементов, из которых может быть построена любая цифровая машина.

Элементы, из которых строится машина, способны реализовать лишь простейшие операции, поэтому возникает вопрос, можно ли представить арифметические действия над числами, записанными в двоичной системе, как некоторую совокупность простых операций только над числами 0 и 1. Это оказывается возможным, если использовать особый вид математических операций, который мы сейчас опишем.

Возьмем числа 0 и 1 и определим для них три новых действия.

Правило для **логического сложения** (логическое «и»): сумма будет равна единице, если хотя бы одно из слагаемых равно единице, **и равна нулю в остальных случаях,** то есть: $0+0=0$; $1+0=0+1=1$; $1+1=1$. Здесь знак плюс не следует смешивать с плюсом в арифметике или в обычной алгебре, так же как далее не следует смешивать знак умножения в правиле для логического умножения.

Правило для **логического умножения** (логическое «или»): произведение равно нулю, если хотя бы один из сомножителей равен нулю, и равно единице, если все сомножители равны единицам, то есть: $0 \times 0 = 1 \times 0 = 0 \times 1 = 0$; $1 \times 1 = 1$.

Напомним правило (стр. 34) для **логического отрицания** (логическое «нет»): логическое отрицание нуля есть единица,

а отрицание единицы есть нуль, то есть $\bar{0}=1$; $\bar{1}=0$. Здесь черточка над цифрами означает действие логического отрицания.

Читатель может сам проверить, что для введенных нами операций **в системе из двух чисел 0 и 1** выполняются следующие законы:

$$a+b=b+a,$$

$$(a+b)+c=a+(b+c),$$

$$\overline{(\bar{a}+\bar{b})}+\overline{(\bar{a}+b)}=a,$$

где a, b и c принимают значения 0 и 1. Такие системы называются **булевыми алгебрами.**

Если теперь попытаться составить такие электрические схемы, чтобы они могли выполнять основные действия булевой алгебры (логическое сложение, логическое умножение и логическое отрицание), то, пользуясь выведенными формулами, можно из этих схем составить новые схемы для выполнения куда более сложных действий. Другими словами, с помощью булевой алгебры можно синтезировать (т. е. составлять) вычислительные схемы дискретного действия и, скажем без пояснений, можно анализировать уже составленные схемы.

Конструирование вычислительной машины обычно начинается с составления так называемых логических схем устройств, в которых изображаются только логические элементы и связи между ними. В конечном итоге логическая схема дает некоторое представление о составе будущей машины, а главное раскрывает ее логические возможности. Часто, еще не зная, как будут выглядеть принципиальные схемы логических элементов, под ними подразумевают некоторые абстрактные схемы, которые способны выполнять основные действия алгебры логики. Так, например, логический элемент «или», изображенный на рис. 7, *а*, выполняет действие логического сложения, логический элемент «и» на рис. 8, *а* выполняет действие логического умножения, а логический элемент «нет» на рис. 9, *а* выполняет действие логического отрицания.

Имея логические схемы устройств, приступают ко второму этапу конструирования — к составлению принципиальных схем логических элементов. Оказывается, что, обладая некоторой изобретательностью, можно составить десятки вариантов релейных и электронных схем, реализующих действия логических элементов. Последовательно замещая элементы логической схемы подходящими электронными схемами, приходят к принципиальной схеме будущего устройства.

Прежде чем рассказывать о принципиальных схемах логических элементов, мы должны условиться о следующих понятиях и допущениях, принятых в дальнейшем:

а) Действующие сигналы в схемах, то есть уровни или импульсы напряжения, изображающие двоичные цифры, называются кодами этих цифр. Описывая устройства ЭЦВМ в следующих разделах, мы для упрощения будем опускать слово «код» и говорить просто «цифра», «знак», «число» и т. д., вместо «код единицы», «код знака» или «код числа».

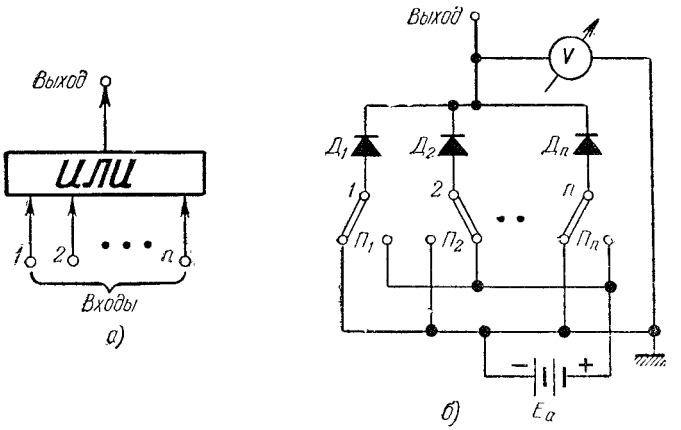


Рис. 7. Логический элемент «или».

Код единицы физически изображается постоянным положительным потенциалом $+E_a$ или положительным импульсом напряжения E_a, где E_a есть э. д. с. источника питания схемы. Код нуля изображается потенциалом земли или отсутствием импульса напряжения. Отрицательный полюс источника заземлен.

б) Чтобы определить код цифры в какой-либо точке схемы, нужно мысленно измерить разность потенциалов между этой точкой и землей. Если разность потенциалов равна E_a, то мы имеем код единицы, если разность потенциалов равна нулю, то имеем код нуля. Подача на вход схемы кодов единицы или нуля, эквивалентна подключению этого входа к полюсу источника $+E_a$ или $-E_a$ (к земле) соответственно.

в) Сопротивление полупроводникового диода в прямом (или проводящем) направлении равно нулю, а сопротивление в обратном (или непроводящем) направлении равно бесконечности. Сопротивление между анодом и катодом проводящей лампы

равно нулю, а сопротивление непроводящей лампы равно бесконечности.

Обратим внимание на то, что принятые нами условности и допущения весьма помогут пониманию других разделов и о них не следует забывать.

Итак, по рис. 7, *б* рассмотрим один из вариантов принципиальных схем, реализующих логический элемент «или». В общем случае схема имеет n входов и один выход. Каждый из входов соединен с выходом посредством диода в прямом направлении. Переключатели $П_1, П_2, \ldots, П_n$ в левом положении осуществляют подачу на соответствующие входы кодов нулей, а в правом положении кодов единиц. К выходу схемы

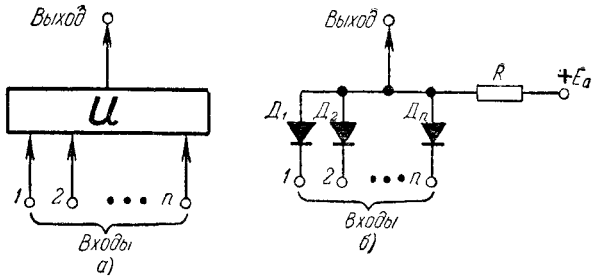


Рис. 8. Логический элемент «и».

подключен вольтметр, которым можно обнаружить код единицы или нуля. Схема работает так, что если на все входы подать коды нулей (переключатели в левом положении), то и на выходе вольтметр обнаружит код нуля. Если на один или более входов подать код единицы (на рисунке переключатель $П_2$ в правом положении), то на выходе вольтметр обнаружит код единицы. Произойдет это потому, что диод $Д_2$ в прямом направлении имеет сопротивление, равное нулю и вход 2 окажется накоротко замкнутым с выходом. Как это легко видеть, схема «или» является по сути дела схемой для собирания кодов единиц, поступающих из многих каналов и поэтому иногда называется собирательной.

На рис. 8, *б* приведен вариант принципиальной схемы, реализующей логический элемент «и». Здесь и на других схемах мы для упрощения больше не будем изображать переключатели для подачи кодов, вольтметр и источники питания, которые всегда можно представить себе мысленно. Схема «и» также имеет в общем случае n входов и один выход,

однако в отличие от схемы «или» диоды подключены к выходу положительными электродами. Кроме того, к выходу постоянно подключено питание $+E_a$ через сопротивление R. Если на все входы одновременно подать коды единиц, то диоды $Д_1$, $Д_2$, ..., $Д_n$ окажутся запертыми, т. е. не будут проводить ток. Ток не будет течь и через сопротивление R, поэтому на выходе схемы появится напряжение $+E_a$ или, как это было условлено, код единицы. Достаточно хотя бы на один из входов, например первый, подать код нуля, как на выходе тотчас появится код нуля по той причине, что от $+E_a$ через сопротивление R и диод, $Д_1$, потечет ток на землю. Так как сопротивление диода в прямом направлении равно нулю, то выход электрически будет закорочен с первым входом, на который подан код нуля. Все напряжение E_a упадет на сопротивлении R. Итак, оказывается, что схема «и» выдает код единицы только в случае одновременного совпадения кодов единиц на всех входах и поэтому иногда называется схемой совпадения. Схему «и» с двумя входами часто еще называют вентилем по той причине, что один из ее входов можно использовать для управления передачей кода по другому входу. Запрет передачи осуществляется посылкой на управляющий вход кода нуля, а разрешение — посылкой кода единицы.

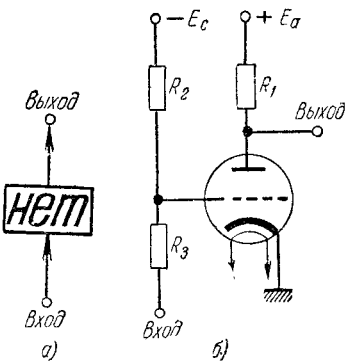


Рис. 9. Логический элемент «нет».

На рис. 9, *б* приведен вариант принципиальной схемы, реализующий логику «нет», т. е. отрицания. Схема представляет собой усилитель постоянного тока на электронной вакуумной лампе триода. Сопротивление R_1 служит здесь анодной нагрузкой лампы, через сопротивление R_2 на сетку лампы подается постоянное отрицательное напряжение смещения $-E_c$ (положительный полюс источника E_c заземлен), а через сопротивление R_3 на сетку поступает входной сигнал. Выходом является анод триода. Сопротивления R_2 и R_3 подбираются с таким расчетом, чтобы при подаче на вход кода единицы лампа непременно была отперта (проводила ток), а при подаче

кода нуля была заперта (не проводила ток). В первом случае на сетке относительно катода образуется положительный потенциал **и лампа открывается,** а во втором случае на сетке образуется отрицательный потенциал и лампа запирается. Если лампа будет открыта, то от $+E_a$ через сопротивление R_1 и лампу на землю потечет ток. Так как по нашему условию сопротивление открытой лампы равно нулю, то ее анод будет как бы закорочен с катодом и выдаст код нуля. Вспомнив, что при этом на вход был подан код единицы, мы и получим логику ее отрицания. Если на вход подать код нуля, то триод окажется в запертом состоянии, что равносильно

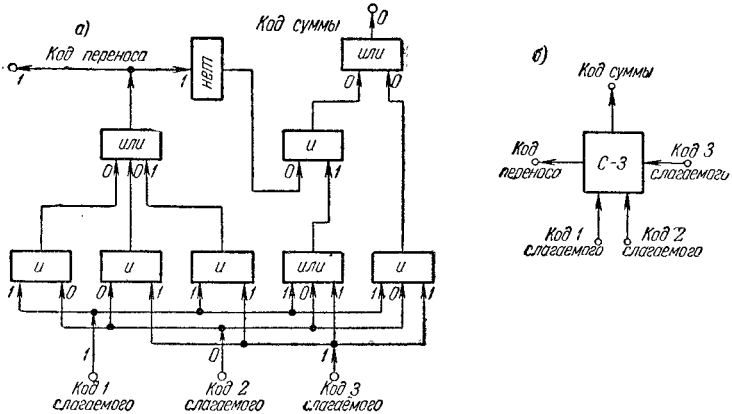


Рис. 10. Логическая схема одноразрядного сумматора.

обрыву цепи анод — катод. Очевидно, что в этом случае на выходе появится напряжение $+E_a$ или код единицы, чем и выполнится логика отрицания нуля.

Примером схемы, составленной из логических элементов, может служить схема одноразрядного сумматора на три слагаемых, приведенная на рис. 10, *а*. Сумматор является наиболее важной частью арифметического устройства ЭЦВМ, в котором собственно и производится арифметическое сложение двоичных цифр. На рис. 10, *б* приведено условное изображение сумматора, где С-3 означает «сумматор на три слагаемых». Согласно правилу арифметического сложения все возможные сочетания цифр на входах сумматора можно свести к табл. 2. Возьмем любую, допустим, седьмую слева вертикальную колонку

В ней представлено сложение следующих трех слагаемых:

$$\begin{array}{r} 1 \\ +0 \\ \quad 1 \\ \hline 1\ 0 \end{array}.$$

На всех входах и выходах логических элементов схемы сумматора (рис. 10, *а*) расписаны значения кодов именно для этого примера. Как это видно из рисунка, в результате взаимодействия логических элементов схемы значения переноса и суммы совпадают с табличными. Читателю представляется

Таблица 2

Входы	Код 1-го слагаемого	0	1	0	0	1	0	1	1
	Код 2-го слагаемого	0	0	1	0	1	1	0	1
	Код 3-го слагаемого	0	0	0	1	0	1	1	1
Выходы	Код суммы	0	1	1	1	0	0	0	1
	Код переноса	0	0	0	0	1	1	1	1

возможность задаться любой комбинацией кодов на входах сумматора, затем, помня о логике действий схем «и», «или» и «нет», расписать последовательно от входов к выходам промежуточные значения кодов и прийти к результату. Описанный сумматор относится к типу кодово-позиционных схем, которые отличаются тем, что не содержат запоминающих элементов. Кодово-позиционные схемы после выполнения действий сами возвращаются в исходное состояние подобно отпущенной пружине. Другие типы схем, счетно-импульсные, содержат запоминающие элементы и хранят на выходах результат предыдущего действия до тех пор, пока на входы не поступит следующая комбинация кодов или не будет сделан преднамеренный сброс схемы в исходное состояние. Работа таких схем подобна действию пружины с защелкой. На рис. 11 для примера приведена логическая схема одного разряда счетно-импульсного двоичного триггерного счетчика.

Самое существенное в схеме триггера заключается в наличии так называемых обратных связей между правой и левой парами логических элементов (цепи обратной связи изображены на рисунке перекрещивающимися линиями). Благодаря обратным связям обеспечивается существование двух устойчивых состояний схемы. Пусть, например, на выходе *С* имеется код единицы. По цепи обратной связи этот код непременно поступит на вход элемента «или$_1$», и далее на вход элемента «нет$_1$». Элемент «нет$_1$» выдает отрицание единицы, т. е. код нуля, который в свою очередь поступит через элемент «или$_2$» на вход элемента «нет$_2$». Код нуля тем самым поддержит состояние элемента «нет$_2$». Если теперь на вход *Б* послать короткий положительный импульс сброса, т.е. код единицы, то элемент «нет$_2$» тотчас выдаст код нуля. Появление же нуля на входе элемента «или$_1$», а следовательно и на входе «нет$_1$», вызовет появление единицы на выходе последнего. Эта единица не замедлит поступить по цепи обратной связи на вход элемента «нет$_2$», чем поддержит и закрепит его предыдущее кратковременное состояние от действия импульса сброса.

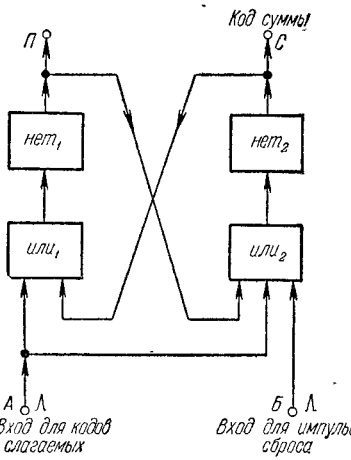


Рис. 11. Логическая схема триггера.

Коды слагаемых подают на вход счетчика *А* в виде коротких положительных импульсов, следующих друг за другом. В исходное состояние схема приводится положительным импульсом сброса, поданным на вход *Б*. При этом на выходе *С* (сумма) появится код нуля, а на выходе *П* (перенос) — код единицы. В целом схема окажется в устойчивом состоянии, которое обеспечивается наличием описанных взаимных обратных связей между входами и выходами логических элементов «нет$_1$» и «нет$_2$». Если теперь на вход *А* (счетный вход) поступит первый импульс слагаемого, то он не может изменить состояние элемента «нет$_2$», но зато изменит состояние элемента «нет$_1$», и на его выходе появится код нуля. В следующий момент времени, когда исчезнет импульс слагаемого, вступят в действие обратные связи и приведут схему в другое, тоже

стабильное состояние. При этом на выходе *П* окажется код нуля, а на выходе *С* код единицы, т. е. будет зафиксирована первая сумма. Импульс следующего слагаемого воздействует уже на элемент «нет$_2$» и возвратит схему в исходное состояние, характеризующееся тем, что по выходу *С* выдается код суммы нуля, а по выходу *П* код переноса единицы.

Таким образом, каждый раз, когда триггер содержит единицу и к ней прибавляется новая единица, на выходе *П* происходит изменение кода с нуля на единицу, что и принимается за сигнал переноса в следующий старший разряд.

2. Арифметическое устройство

Арифметическое устройство вычислительной машины предназначено для выполнения арифметических и логических операций. К числу последних относятся уже знакомые нам операции логического сложения, логического умножения, операция сдвига числа вправо или влево, операция сравнения чисел и т. д. Чем больше разнообразие операций, которые выполняет арифметическое устройство, тем бо́льшими возможностями располагает программист и тем легче и проще составляются программы решения задач.

По принципу работы арифметические устройства делятся на устройства последовательного и параллельного действия. В устройствах последовательного действия сложение чисел выполняется с помощью всего лишь одного разряда сумматора. Числа, изображенные группой импульсов последовательно распределенных во времени, одновременно поступают из запоминающего устройства на входы сумматора. Первыми поступают младшие разряды чисел. Очередные цифры двух чисел складываются каждый раз в сумматоре, а перенос временно запоминается и прибавляется в следующем цикле к сумме новой пары цифр. Такой принцип работы не может дать большой скорости вычислений, но зато позволяет получить простое и недорогое арифметическое устройство.

На рис. 12 схематически показано последовательное арифметическое устройство и даны изображения слагаемых 9, 13 и суммы 22 с помощью последовательных серий импульсов. Положительные импульсы изображают единицы, а прочерки — нули. Сумма выдается сумматором также последовательно — разряд за разрядом.

В арифметическом устройстве параллельного действия два числа складываются всеми своими разрядами одновременно.

Это дает возможность получить большую скорость работы за счет значительного усложнения устройства. Если в арифметическом устройстве последовательного действия требуется только один разряд сумматора, то в устройстве параллельного действия сумматоров нужно столько, сколько разрядов занимает наибольшее число.

С точки зрения самой техники выполнения схем арифметических устройств последние делятся на импульсные и кодово-позиционные, или просто позиционные. Простейшим примером импульсной схемы может служить триггер (рис. 11). Примером позиционной схемы может служить сумматор, приведенный на рис. 10.

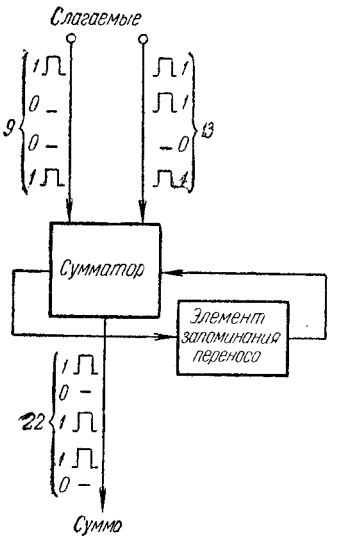


Рис. 12. Общая блок-схема запоминающего устройства.

Мы опишем наиболее простые для понимания кодово-позиционные арифметические устройства. Будут рассмотрены устройства параллельного действия, которые могут выполнять операции сложения, вычитания и умножения над числами с фиксированной запятой.

Для арифметического устройства сложения и вычитания нужны две основные схемы: схема сумматора и схема для получения обратного кода отрицательного числа, которую назовем преобразователем. Если схема сумматора нам уже известна, то логическая схема преобразователя, приведенная на рис. 13, требует пояснений. Из табл. 3, куда сведена логика работы преобразователя, и из логической схемы преобразователя видно, что если на вход преобразователя поступает цифра записи положительного числа, знак которого изображается нулем, то схема выдает эту цифру без изменений. Если же на вход поступает цифра записи отрицательного числа, знак которого изображается единицей, то преобразователь изменяет эту цифру на обратную, т. е. с единицы на нуль и с нуля на единицу, чем и достигается получение обратного кода. На том же рисунке слева дано условное изображение одного разряда преобразователя.

Таблица 3

Входы	Код цифры (разряд числа)	0	1	0	1
	Код (знака числа)	0 (+)	0 (+)	1 (−)	1 (−)
Выход	Результат преобразования	0	1	1	0

Из сумматора и преобразователя можно уже составить устройство для сложения и вычитания. Логическая схема

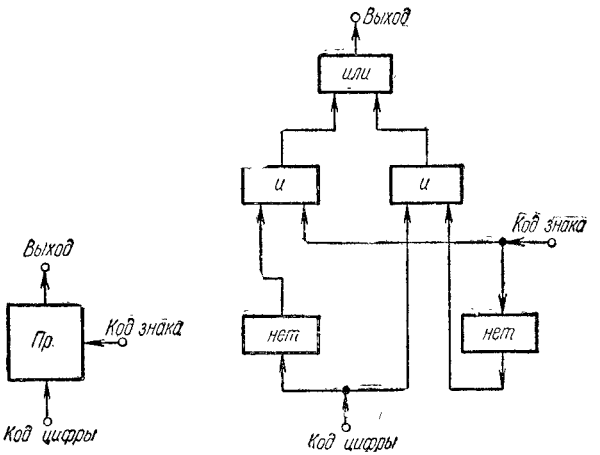


Рис. 13. Логическая схема преобразователя.

такого устройства приведена на рис. 14. На входы цепочек преобразователей *Пр. 1* и *Пр. 2* из запоминающего устройства подаются первое и второе слагаемые. В цепочке сумматоров *С*-3 осуществляется сложение кодов. Знак суммы получается двоичным суммированием знаков слагаемых в сумматоре *С*-3 (*5*) с учетом переноса из дополнительного разряда. Дополнительный разряд сумматора *С*-3 (*4*) нужен на случай, когда сумма займет больше разрядов, чем имеет каждое из слагаемых. Для передачи единицы циклического переноса выход старшего разряда цепочки сумматоров соединен с входом младшего разряда. Цепочка преобразователей *Пр. 3* необходима для обратного преобразования суммы, если она получится в обратном

коде, т. е. будет отрицательна. Результат операции передается из арифметического устройства в запоминающее устройство.

Поскольку алгебраическое вычитание чисел отличается от сложения только изменением знака второго слагаемого, то с помощью одного и того же устройства можно выполнять и операцию вычитания. Для этой цели в устройство введен преобразователь знака *Пр. зн.* На один из его входов подается

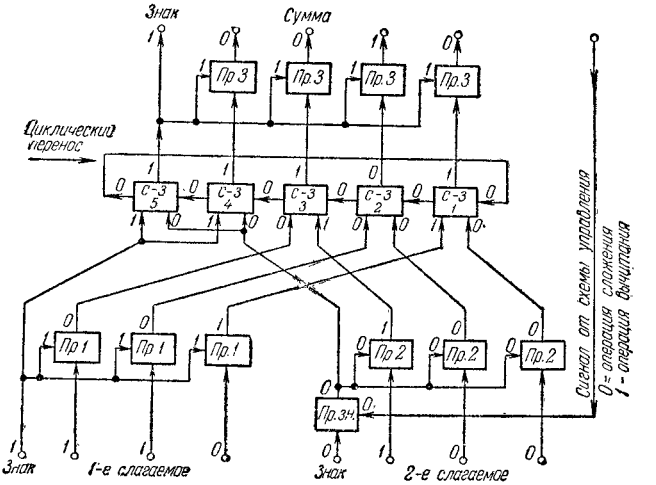


Рис. 14. Логическая схема устройства для сложения и вычитания

знак второго слагаемого, а на другой вход от схемы управления арифметического устройства подается нуль, если нужно выполнить операцию сложения, и единица, если нужно выполнить операцию вычитания.

Для того чтобы понять работу схемы, достаточно с «ее помощью» решить несколько примеров. Возьмите два любых трехразрядных двоичных числа и проследите по схеме, как получается их сумма. Сложите, например, $(+5)+(+6)$, $(+7)+(-3)$, $(+2)+(-6)$, $(-4)+(-5)$ или найдите разность $(-1)-(-3)$, $(+7)-(+7)$.

В качестве примера на элементах схемы расписаны начальные, промежуточные и окончательные результаты операции алгебраического сложения двух чисел: $(-6)+(+4)=-2$ или в двоичной системе:

$$(-110)+(+100)=(-010).$$

Путем значительного усложнения можно составить схему устройства для действий над числами с плавающей запятой. При этом рассмотренная схема может быть использована для сложения мантисс, если добавить к ней сдвигатель вправо и сдвигатель нормализации. С помощью первого осуществляется сдвиг вправо мантиссы меньшего числа, что нужно для уравнивания порядков, а с помощью второго — нормализация результата. Для нахождения разности порядков описанная схема подходит без всяких изменений или добавлений, если на правый вход преобразователя знака всегда подавать единицу, а на входы цепочек преобразователей *Пр. 1* и *Пр. 2* подавать порядки и их знаки (полная аналогия с операцией вычитания второго слагаемого из первого!).

Нетрудно заметить, что логическая схема кодово-позиционного арифметического устройства отображает естественный ход операций сложения, вычитания и, скажем наперед, операции умножения. Это положение существенно облегчает конструирование, но устройства получаются очень громоздкими по количеству элементов. Кодово-позиционные схемы имеют примерно в три раза больше элементов, чем импульсные.

Рассматривая следующий пример умножения двух чисел:

$$
\begin{array}{rll}
 & \times 1101 & \\
 & \times 1011 & \\
\hline
 & 1101 & (а) \\
 & 1101 & (б) \\
 & 0000 & (в) \\
 & 1101 & (г) \\
\hline
 & 10001111 &
\end{array}
$$

частные произведения

можно прийти к выводу, что операция умножения сводится к простому суммированию частных произведений, сдвинутых друг относительно друга на один разряд влево. Сами же частные произведения равны множимому, если соответствующий разряд множителя равен единице и нулю, если разряд множителя равен нулю. Обратимся к рис. 15, где приведена логическая схема множительного устройства. В состав схемы входят четыре группы логических элементов «и», выполняющих роль вентилей множимого. Частные произведения определяются соответствующими цифрами множителя, которые управляют работой вентилей. Если цифра множимого есть единица, то группа вентилей открыта и на сумматоры пропускается множимое, если эта цифра есть нуль, то вентили закрыты и частное произ-

ведение равно нулю. Для получения произведения частные произведения суммируются в трех цепочках сумматоров. Две первые цепочки дают суммы первой $(a+б)$ и второй $(в+г)$ пары частных произведений, а последняя цепочка дает сумму $(a+б)+(в+г)$ частных произведений, то есть результат умножения. Сдвиг частных произведений достигается определенными и постоянными соединениями в схеме. Работу устройства можно иллюстрировать примером умножения двух чисел, промежуточные результаты которого написаны на элементах схемы. Попытайтесь самостоятельно задаться любыми двумя четырехразрядными числами и проследить по схеме, как выполняется операция умножения. Нужно только помнить, что запятая у чисел должна быть фиксирована.

Из устройства с фиксированной запятой легко получить множитель для чисел с плавающей запятой. Для этого нужно добавить: алгебраический сумматор порядков, схема которого ничем не отличается от схемы устройства сложения, несложную схему для получения знака произведения и схему для нормализации произведения, на случай если справа от запятой появится нуль.

Описанные принципы используются в арифметическом устройстве машины «Стрела». В таких машинах, как БЭСМ, «Урал» и в большинстве машин иностранных марок используются импульсные принципы действия, дающие более экономичные устройства.

3. Запоминающее устройство

Запоминающее устройство **ЭЦВМ** удобно сравнить с памятью человека. Оценивая память человека, мы обычно прибегаем к таким критериям, как скорость, с которой человек может вспомнить то или иное событие, и объем сведений о событиях, хранящихся вообще в его памяти. Если предположить, что все сведения, или, как мы условимся их называть, вся *информация* разложена в памяти по «полочкам», то легко согласиться со следующим. Ежедневно в быту человек пользуется лишь небольшой группой «полочек», содержимое которых составляет его обиходный словарь. Этот словарь, как правило, не превышает нескольких сотен слов (названия окружающих предметов, описание простых взаимоотношений и т. д.). Условно можно считать, что такие слова расположены на *ближних* «полочках» памяти, и человек без всяких затруднений обращается к ним в любое время. Эту небольшую группу ближних

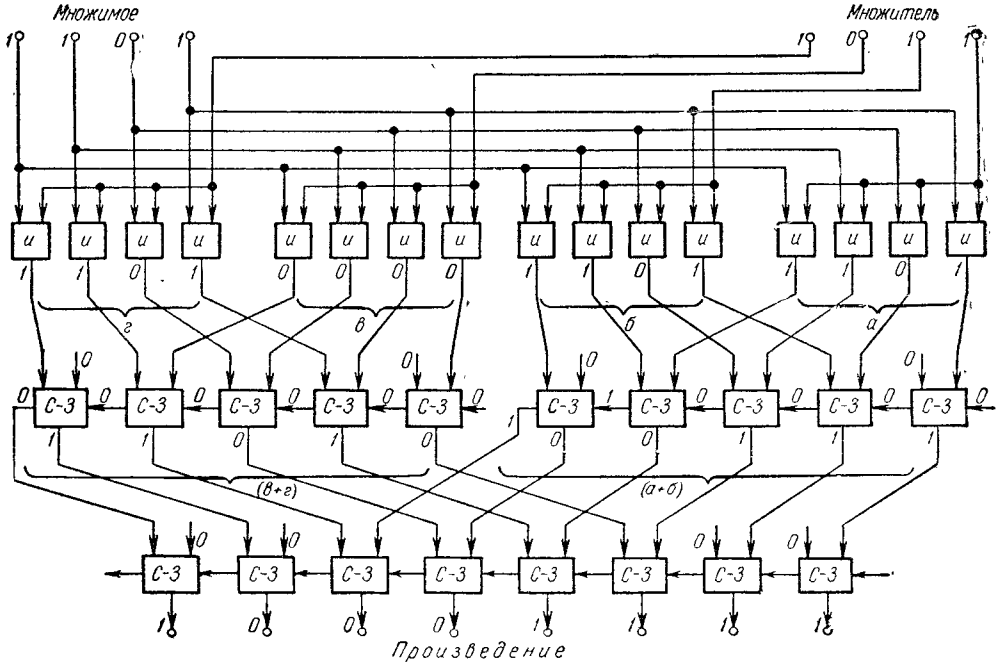


Рис. 15. Логическая схема устройства для умножения.

«полочек» можно охарактеризовать как быстродействующую часть памяти и назвать ее, пользуясь терминологией вычислительной техники, оперативной или внутренней.

Другая часть информации разложена по *дальним* «полочкам» и используется сравнительно редко (подробности содержания давно прочитанной книги, события давних лет и т. д.). Обращение к этой части информации уже требует большей затраты времени и энергии. Каждый из нас, наверно, знает более 30 фамилий писателей и поэтов, но попробуйте без остановок перечислить хотя бы 15. Оказывается, что это не всегда удается сделать. Но если имеется достаточно времени, то, «покопавшись» в памяти, можно вспомнить и остальные фамилии. Это позволяет нам назвать вторую группу «полочек» медленно действующей частью памяти. Объем информации, хранящейся в ней, значительно больший, чем в быстродействующей части. Пользуясь опять терминологией вычислительной техники, назовем вторую часть памяти медленной или внешней.

Оба вида памяти имеют общую особенность, которую можно выразить словом активность. Смысл активности заключается в том, что мы можем обращаться к памяти не только за нужной информацией, но и запоминать в ней по мере необходимости новую.

Однако мы забыли еще об одном виде памяти, которую назовем пассивной. К ней относится необъятная библиотека книг, документов, учебников, справочников и все то, что хранит знания многих поколений человечества. Термин *пассивная* оправдывается здесь дважды: пока книгу не читают, она не более как пассивный груз на книжной полке, и второе, на уже отпечатанные страницы книги нового текста не печатают. Другими словами, книгу можно только читать, т. е. извлекать из нее информацию, а запоминать с ее помощью новую нельзя.

В заключение осталось сказать, что, разбивая память человека на «полочки», мы это сделали не случайно. Конечно, никто из нас не задумывался над тем, в каком месте мозга запоминается то или иное новое слово и откуда «извлекается» уже однажды услышанное слово. Нас сейчас не интересует этот вопрос, но нам важно знать, в какое место запоминающего устройства **ЭЦВМ** записывается очередной результат вычислений. Ведь если это будет неизвестным, то мы не сможем найти его и прочитать, когда в этом возникнет необходимость. Здесь-то и приходят на помощь «полочки». Все запоминающее

устройство разбивается на «полочки» или, как они называются, ячейки. «Размер» ячейки рассчитан на запоминание одного числа или команды. Все ячейки имеют сквозную нумерацию, начиная с первой, а каждый номер называется адресом. Таким образом, если теперь нужно записать в запоминающее устройство число, то мы должны знать номер ячейки или адрес ячейки, куда это число записывается. Зная же адрес, всегда можно по нему найти нужную ячейку и прочитать то, что было туда записано ранее.

Запоминающие устройства цифровых машин предназначены для запоминания, хранения и выдачи команд программы, различных вспомогательных данных, промежуточных и конечных результатов. Различают три вида памяти: внутреннюю, внешнюю и пассивную. Внутренняя память хранит ту часть программы, которая непосредственно участвует в вычислениях. Внешняя память хранит остальную часть программы. Пассивная память хранит различные константы, таблицы и все то, что может одинаково пригодиться при решении многих задач. В управляющих машинах пассивная память может хранить и программу вычислений.

По мере того как заканчивается выполнение программы, хранящейся во внутренней памяти, ненужная ее часть либо стирается, либо переносится во внешнюю память, а на освободившееся место из внешней памяти переносится новая часть программы. Как правило, внешняя память записывает, считывает или стирает сразу группу чисел, а внутренняя память к тому же может записывать, считывать или стирать любое индивидуальное число. Пассивная память позволяет выполнять только считывание.

Основными техническими характеристиками памяти являются скорость ее работы и емкость. Быстродействие машины в значительной мере зависит от того, какова скорость обмена информацией между внутренней памятью и арифметическим устройством, так как все исходные данные для арифметических действий берутся из внутренней памяти. Поэтому к внутренней памяти прежде всего предъявляется требование большой скорости работы при сравнительно малой ее емкости. Внешняя память должна обладать возможно большей емкостью, и малые скорости работы здесь допустимы из-за относительно редкого участия в процессе вычислений. Емкость пассивной памяти может меняться в зависимости от назначения машины в самых широких пределах, однако скорость ее работы бывает такой же, как и у внутренней памяти. Последнее объясняется тем,

что содержимое ячеек пассивной памяти принимает участие в арифметических действиях и, более того, пассивная память может хранить отдельные стандартные «куски» программы. К таким стандартным программам могут относиться: программы вычисления тригонометрических функций, программы перевода чисел из десятичной системы счисления в двоичную и т. д.

По рис. 16 рассмотрим в самом общем виде порядок, в котором производится запись или чтение чисел в запоминающем устройстве. Устройство состоит из блока запоминания и

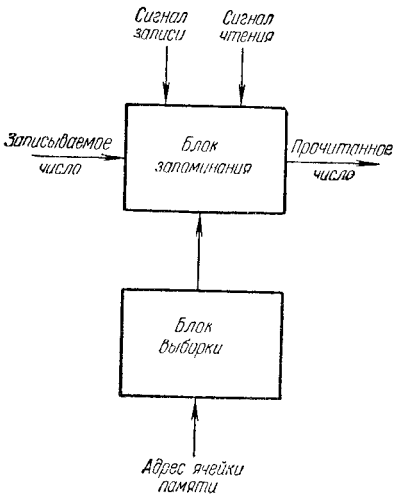


Рис. 16. Общая блок-схема запоминающего устройства.

блока выборки. Блок запоминания содержит элементы, с помощью которых непосредственно осуществляется запоминание чисел (магнитная лента, ферритовые тороиды и т. д.). Блок выборки представляет собой схему, которая по заданному адресу находит нужную группу запоминающих элементов в блоке запоминания. Работа устройства протекает следующим образом. На вход блока выборки от устройства управления машины поступает адрес ячейки. По указанному адресу блок выборки находит группу запоминающих элементов (или ячейку) в блоке запоминания. Если в данную ячейку нужно записать число, то от управления машины в блок запоминания поступает сигнал записи и число запоминается избранной группой элементов. Если нужно прочитать содержимое ячейки, то от

управления машины поступает сигнал чтения. Если содержимое какой-либо ячейки нужно стереть, то в нее попросту записывают нуль. Заметим здесь, что при записи в ячейку памяти нового числа старое автоматически стирается.

Не имея возможности останавливаться на принципиальных схемах запоминающих устройств, мы тем не менее познакомимся с физикой запоминания двоичных цифр с помощью различных элементов.

В повседневной жизни мы то и дело встречаемся с разнообразными простейшими запоминающими устройствами или с физическими явлениями, очень близкими по своему существу с запоминанием той или иной информации.

Вы поворачиваете выключатель, и в комнате загорается свет. Информация «включить свет» *запоминается* механизмом выключателя, и он останется в новом устойчивом состоянии, пока не получит другую информацию — «выключить свет». Оба устойчивых состояния выключателя можно обозначить двоичными цифрами. Свет включен — единица, свет выключен — нуль.

Перед подъемом на лифте вы нажимаете кнопку требуемого этажа. Ваше желание в виде электрических импульсов передается по проводам в машинное отделение лифта, где *запоминается* и исполняется с помощью группы электромагнитных реле.

Музыка записывается на патефонную пластинку и она *запоминает* «музыкальную информацию» неопределенно долго. Этот пример хорошо иллюстрирует пассивную память.

Наконец, музыку можно записать с помощью магнитофона на магнитную ленту и лента добросовестно *запомнит* ее. Надоевшую мелодию можно стереть с магнитной ленты и на то же место записать новую. Пример с магнитофоном иллюстрирует активную память.

Если в перечисленных примерах факт запоминания в какой-то степени очевиден, то утверждение, что обычный воздух тоже может запоминать, потребует пояснений. Известно, что свет от молнии практически мгновенно достигает нашего зрения, а гром мы слышим спустя многие секунды после молнии. Дело, конечно, в том, скажете вы, что звук распространяется намного медленнее, чем свет. Это действительно так. Но действительно и то, что воздух *запомнил* звуковые колебания и спустя некоторое время донес их до нашего слуха. Если на расстоянии нескольких сот метров от наблюдателя работает отбойный молоток, то в воздушной среде по направлению от молотка к наблюдателю может одновременно запоминаться

несколько ударов. Легко сообразить, что каждый удар может изображать двоичную единицу, а пропуск удара — нуль. Именно на этом явлении распространения звука в среде с некоторой конечной скоростью и основаны запоминающие устройства с так называемыми ультразвуковыми или акустическими линиями. Подобные запоминающие устройства получили наибольшее распространение в 1946—1953 гг. Схематическое изображение одной из разновидностей акустических линий, в

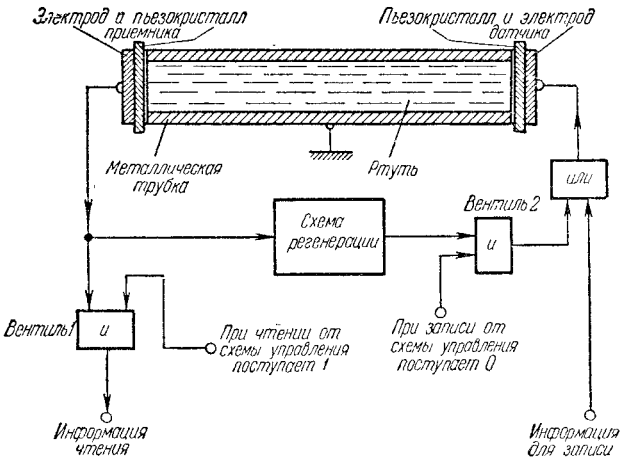


Рис. 17. Схема памяти на акустической ультразвуковой линии.

которой в качестве запоминающей среды используется ртуть, вода или другая жидкость, показано на рис. 17. Поясним устройство линии.

Столбик ртути заключен в металлическую трубку. С одного конца трубка имеет пьезоэлектрический датчик, который преобразует электрические импульсы в акустические, а с другого конца — пьезоэлектрический приемник, преобразующий акустические импульсы в электрические. Сущность запоминания сводится к тому, что акустические импульсы, излучаемые датчиком и изображающие двоичные цифры, распространяются в среде с конечной скоростью *). Время распространения импульсов от датчика к приемнику и является временем их хранения в линии. Внешняя цепь трубки содержит схему регенерации, которая

*) Скорость распространения звука во многих жидкостях составляет около 1500 *м/сек*.

принимает электрические импульсы от приемника, усиливает их и синхронно посылает в датчик. Таким образом, однажды введенная в трубку информация может восстанавливаться и неопределенно долго циркулировать по замкнутому кольцу, в чем и заключается, собственно, регенерация. В момент считывания вентиль *1* направляет импульсы от приемника в арифметическое или другое устройство. При записи вентиль 2 перекрывает замкнутое кольцо, чем достигается стирание ранее записанной информации, а новая информация направляется в датчик и записывается на место старой.

В зависимости от конструкции трубка может хранить одновременно 1000 двоичных цифр. Время записи или чтения одной цифры равно примерно микросекунде. Следовательно, на запись или чтение 20-разрядного числа будет затрачено около 20 *мксек*. Однако такое время будет затрачено только в том случае, если к требуемому моменту нужное число подходит к вентилю *1*. Если же число не успело подойти, то приходится тратить время на его ожидание, что замедляет общий темп работы запоминающего устройства. В среднем же время чтения, включая время ожидания, составляет около тысячной доли секунды.

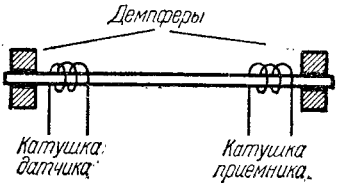


Рис. 18. Магнитострикционная линия задержки.

В 1951 г. впервые для запоминания двоичных чисел была использована акустическая линия из твердой среды. Опишем кратко принцип ее работы. Тонкая никелевая проволока длиной в несколько метров (рис. 18) имеет на одном конце катушку датчика, а на другом катушку приемника. При посылке в катушку датчика импульса тока возникает магнитное поле, которое вызывает в проволоке магнитострикционный эффект. Действие магнитострикционного эффекта проявляется в том, что в проволоке возникает упругая механическая деформация, которая в виде акустической волны распространяется в противоположные от катушки стороны с большой скоростью. Волна, двигающаяся к ближнему концу проволоки, подавляется резиновым демпфером, а другая волна двигается к катушке приемника, в которой и наводит э.д.с. Время движения волны от датчика к приемнику является временем ее хранения в линии. Остальная часть устройства мало отличается от только что описанного на линиях с жидкой средой.

Иные принципы использованы в запоминающих устройствах на магнитной ленте или магнитном барабане. Запоминающие устройства на магнитной ленте применяются преимущественно для внешней памяти, а на магнитном барабане — как для внутренней, так и для внешней памяти. Оба типа памяти используют принцип запоминания, ничем не отличающийся от запоминания музыки и речи на обычном магнитофоне. Только вместо звуковых колебаний на магнитную ленту или барабан с магнитным покрытием записываются двоичные цифры в виде намагниченных участков.

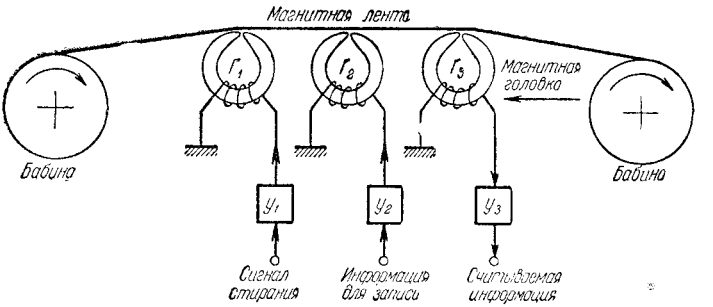


Рис. 19. Схема устройства памяти на магнитной ленте.

На рис. 19 приведено схематическое изображение устройства памяти на магнитной ленте. При работе лента перематывается с одной катушки на другую и движется, касаясь читающей $Г_3$, записывающей $Г_2$ и стирающей $Г_1$ магнитных головок. Головка представляет собой подковообразный сердечник из материала с большой магнитной проницаемостью. На сердечнике имеется обмотка, подключенная к своему усилителю ($У_1$, $У_2$, $У_3$). Если нужно записать на ленту двоичную единицу, то усилитель $У_2$ посылает в обмотку головки $Г_2$ импульс тока. В сердечнике головки и между его полюсами возникает магнитное поле, которое и намагничивает участок движущейся ленты. Если участок ленты не намагничен, то считают, что там записан нуль. При чтении информации намагниченный участок ленты проходит под полюсами читающей головки $Г_3$ и возбуждает в ее сердечнике резко нарастающий и спадающий магнитный поток. Изменение магнитного потока индуцирует в обмотке головки э.д.с., которая затем усиливается усилителем $У_3$ и передается по назначению дальше. Если под читающей головкой проходит не намагниченный участок ленты, то, естественно, в ее обмотке не индуцируется э.д.с., что будет изо-

бражать прочитанный нуль. Для стирания ненужной информации на обмотку головки $Г_1$ подается переменный ток высокой частоты и движущаяся под ней лента размагничивается. Описанный способ записи, чтения и стирания не единственный. Можно, например, стирать ленту, равномерно намагничивая ее одной полярностью, а запись единиц вести обратной полярностью. Другой способ состоит в том, что при записи каждой следующей единицы полярность намагничивания ленты меняется на

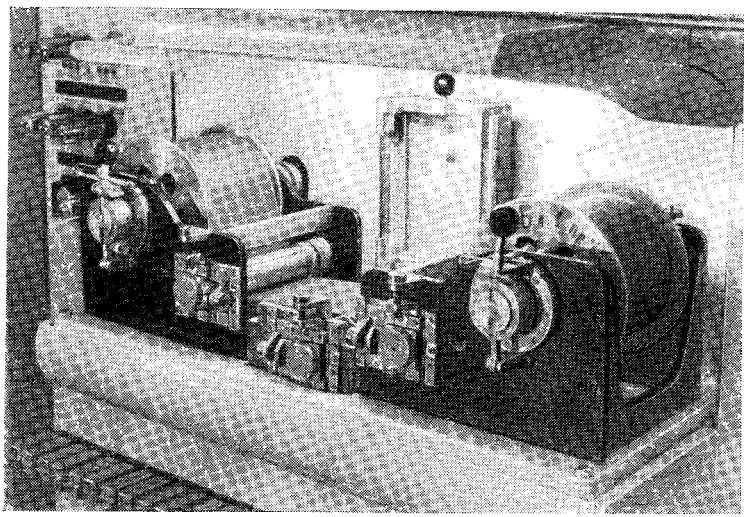


Рис. 20. Лентопротяжный блок машины «Стрела».

обратную, а при записи нулей лента намагничивается той полярностью, в которой была записана последняя единица. Несмотря на некоторую сложность электрической схемы, последний способ выгодно отличается от остальных тем, что на 1 *мм* длины ленты можно записывать 40—60 знаков. Плотность же записи для первого описанного способа не превышает 10—15 знаков на 1 *мм* длины ленты.

Емкость внешней памяти на магнитной ленте практически не ограничена и зависит только от ее конструкции (длина ленты, количество сменных бобин с лентой, плотность записи и т. д.). Внешняя память на магнитной ленте машины «Стрела», например, позволяет хранить одновременно до 500 000 чисел и команд. На рис. 20 приведена фотография протяжного блока

магнитной ленты машины «Стрела». Под открытой крышкой видна магнитная лента. Магнитные головки скрыты под лентой.

Скорость работы запоминающих устройств на магнитной ленте сравнительно невелика и зависит: от конструкции лентопротяжного блока, от механической прочности ленты и от принципа записи или чтения. В лучшем случае одна магнитная головка может записывать или считывать несколько тысяч цифр в секунду. Однако количество информации, которое может храниться на ленте, настолько велико, что подобные запоминающие устройства нашли самое широкое применение в вычислительной технике.

Магнитный барабан делают из легкого и прочного немагнитного сплава. На поверхность барабана наносят тонкую пленку магнитного материала, подобного тому, которым покрыта магнитная лента. Иногда барабан гальванически покрывается слоем магнитного металла, например кобальта. Вдоль образующих барабана, на небольшом расстоянии от его поверхности (30 ÷ 60 микрон), располагаются магнитные головки записи, чтения и стирания. Очень часто в одной головке совмещаются функции записи и чтения. Все остальное, сказанное о принципе действия магнитной ленты, в той же мере относится и к барабану. Вращаясь со скоростью нескольких тысяч оборотов в минуту, магнитный барабан позволяет записывать и читать одно число за сотые доли секунды. Емкость барабана редко превышает 5 ÷ 10 тысяч чисел.

Следующими запоминающими элементами, с которыми предстоит нам познакомиться, являются ферритовые тороидальные сердечники, или просто — ферритовые тороиды.

Если в отверстие соленоида, по которому пропускается постоянный ток, поместить стальной стержень, то он окажется намагниченным с одного конца северным, а с другого южным магнитными полюсами. Поместив стержень в соленоид другой стороной, мы тем самым намагнитим его концы обратными полярностями, что можно проверить стрелкой компаса. Как первое, так и второе состояние намагниченности достаточно устойчивы и могут сохраняться длительное время, если, конечно, вновь не воздействовать на стержень посторонним магнитным полем или не подвергать его ударам. Описанное явление хорошо известно нам и нетрудно догадаться, что одно из магнитных состояний стержня может изображать единицу, а другое нуль. Если намагниченный стержень свернуть в кольцо (тороид), плотно соединив его концы, то он сохранит намагниченность, однако обнаружить ее полярность будет уже сложнее. Для

этого на тороид наматываются две катушки из провода. Через одну катушку кратковременно пропускают ток и, если направление магнитных силовых линий катушки не совпадает с направлением магнитных силовых линий тороида, то последний перемагнитится. Перемагничивание тороида вызывает изменение магнитного потока, который, пронизывая витки второй катушки, индуцирует в ней э.д.с. Появление э.д.с. указывает на то, что произошло перемагничивание тороида. Если э.д.с. не появилась, то, очевидно, магнитное поле первой катушки совпало с магнитным полем тороида. Таким образом, если известно направление магнитных силовых линий первой катушки, то появление или отсутствие э.д.с. во второй катушке подскажет первоначальное магнитное состояние тороида.

Выбирая направление тока в катушке, можно намагничивать тороид заранее известной полярностью.

Однако общее знакомство с магнитными свойствами стального тороида нужно дополнить рядом других сведений, необходимых для уяснения принципов работы запоминающего устройства.

Магнитное состояние тороида может быть охарактеризовано магнитным потоком Φ, который пронизывает сечение тороида. При неизменных внешних условиях магнитный поток остается также неизменным. Но если на тороид воздействовать внешним магнитным полем, например с помощью обмотки на самом тороиде, через которую пропускается ток, то магнитный поток изменится. Новое значение магнитного потока будет зависеть от величины напряженности H магнитного поля обмотки. Если на оси абсцисс отложить значения напряженности H магнитного поля, а на оси ординат — соответствующие значения магнитного потока (индукции), то зависимость магнитного потока от напряженности можно представить в виде кривой (рис. 21, *б*), которая называется петлей гистерезиса. Заметим, что величина H пропорциональна току, протекающему через обмотку, поэтому по абсциссе можно было бы вместо H откладывать значения силы тока I. Изменяя величину и направление тока в обмотке, мы будем тем самым изменять величину и знак напряженности магнитного поля. Последнее вызовет в свою очередь изменение величины направления (знака) магнитного потока в тороиде. Замкнутый характер кривой говорит о повторяемости процесса намагничивания тороида. Так, если напряженность магнитного поля менять в пределах от $-h$ до $+h$, затем от $+h$ до $-h$ и в дальнейшем повторять этот цикл много раз, то магнитный поток будет меняться от $-\Phi_1$ до $+\Phi_2$, от $+\Phi_2$ до $-\Phi_1$ и т. д., повторяясь снова в той же последовательности.

Все магнитные материалы имеют одну очень важную особенность, заключающуюся в том, что переход из одного состояния намагниченности в другое может происходить только по определенному правилу, которое поясняется стрелками на петле гистерезиса. Так, например, чтобы перемагнитить тороид из состояния, изображаемого точкой *а*, в состояние *б*, нужно приложить напряженность магнитного поля, равную $+h$, и затем уменьшить ее до нуля. При этом переход из состояния *а* в состояние *б* произойдет только по стрелке через точки *1*, *2*, *3*.

Если же приложить напряженность не $+h$, а $-h$ и затем сделать ее вновь равной нулю, то тороид перейдет сначала в

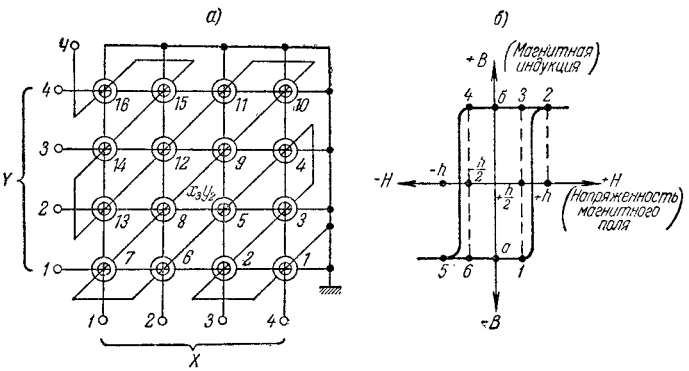


Рис. 21. Схема матрицы ферритового запоминающего устройства.

состояние, обозначенное точкой *5*, после чего опять возвратится в состояние *а*, т. е. перемагничивания не произойдет.

Вторым важным свойством магнитных материалов, но уже теперь не всех, а только тех, которые используются для устройств памяти, является их способность резко переходить из одного состояния намагниченности в другое. Так, например, если тороид находится в состоянии *а*, то напряженность $+h/2$ еще не может произвести перемагничивание. Достаточно немного увеличить напряженность, как произойдет резкое перемагничивание. Петля гистерезиса материала с подобными свойствами приближается по своей форме к прямоугольнику и поэтому называется прямоугольной.

Третье свойство тороидов для устройств памяти должно заключаться в способности долго сохранять одно из двух состояний намагниченности.

Описанные физические явления и были использованы в основе запоминающего устройства на магнитных ферритовых тороидах, которые предложили в 1953 г. американцы Кауихан, Хеинс и Уитни.

Тороиды изготовляются из спрессованных, а затем обожженных особым способом порошкообразных ферритовых материалов. Полученные таким образом ферритовые тороиды, обладают прямоугольной петлей гистерезиса, способностью быстро намагничиваться и долго сохраняют магнитные свойства.

На рис. 21, *а* приведены простейшая схема памяти на ферритовых тороидах и уже знакомая нам петля гистерезиса, характеризующая магнитные свойства тороида. Схема представляет координатную сетку из проводов записи X_1, X_2, X_3, X_4 и Y_1, Y_2, Y_3, Y_4, которые пронизывают тороиды в местах своего пересечения. Кроме того, все тороиды пронизываются одним общим проводом чтения *Ч*. Каждый из проводов, пронизывающих тороид, образует для него один виток обмотки. Если ток в обмотках отсутствует, то напряженность внешнего магнитного поля равна нулю и тороид находится в состоянии *а* или *б*, что очевидно из петли гистерезиса. Условимся, что состояние *а* изображает нуль, а *б* — единицу. Условимся также, что ток одной обмотки, равный $i/2$, создает напряженность магнитного поля $h/2$, которая в зависимости от знака может перевести тороид из состояния *а* или *б* в состояние *1*, *6* или *3*,*4* соответственно. Однако поле напряженностью $h/2$ еще недостаточно для полного перемагничивания тороида и, после прекращения его действия при $h=0$ тороид вернется в исходное состояние *а* и *б*. Чтобы перейти из точки *а* в *б* или из точки *б* в *а* нужно, в первом случае, приложить напряженность $+h$, а во втором $-h$, и после перехода в точки *2* или *5* прекратить действие внешнего поля. Процесс перемагничивания будет идти в первом случае через точки *а*, *1*, *2*, *3*, *б*, а во втором случае через точки *б*, *4*, *5*, *6*, *а*. Проследим теперь за работой схемы на примере одного тороида (X_3, Y_2), предположив, что на всех тороидах в исходном состоянии были записаны нули.

Для записи в избранный тороид единицы, в обмотки X_3 и Y_2 одновременно подаются короткие импульсы тока амплитудой $+i/2$. В остальных обмотках ток отсутствует. Все тороиды, пронизываемые обмотками X_3 и Y_2, окажутся под действием напряженности $+h/2$ и не смогут перейти из состояния *а* в состояние *б*. Но избранный тороид будет находиться под действием суммарного магнитного поля напряженностью $+h$, которое образовано половинными и одинаково направленными

полями обмоток X_3, Y_2. Поле $+h$ переведет тороид из состояния *а* в состояние *б*. Это и будет запись единицы. Состояния остальных тороидов не изменятся. Для записи нуля на те же обмотки X_3, Y_2 подаются одновременно короткие импульсы тока амплитудой $-i/2$. Из всех тороидов, пронизываемых обмотками X_3, Y_2, только один избранный перейдет из состояния *б* в *а*, так как окажется под действием суммарного поля $-h$ обеих обмоток. Остальные тороиды останутся в первоначальном состоянии. Чтобы прочитать запись, на входы обмоток X_3, Y_2 подаются импульсы тока амплитудой $-i/2$, благодаря чему в избранный тороид записывается нуль. Если тороид находился в состоянии *а*, то при чтении его состояние не изменится и, следовательно, не изменится магнитный поток и в обмотке *4* напряжение не появится. Если при чтении тороид перешел из состояния *б* в *а*, то изменение магнитного потока индуцирует в обмотке чтения *Ч* импульс напряжения, который и воспринимается последующими схемами как единица. Специальная схема регенерации временно запоминает прочитанную цифру и после чтения вновь записывает ее на тот же самый тороид.

В приведенном простейшем примере координатная сетка содержит всего 16 тороидов. Для запоминания шестнадцати сорокаразрядных чисел потребовалось бы сорок координатных сеток из 640 тороидов. Для запоминания 10 000 сорокоразрядных чисел потребовалось бы 400 000 тороидов. Однако конструкция ферритовой памяти не получилась бы громоздкой ввиду чрезвычайно небольших размеров тороидов. Так, наиболее распространенные типы тороидов имеют диаметр всего 1—2 *мм*.

Время записи или чтения для ферритового запоминающего устройства исключительно мало и составляет всего несколько микросекунд, поэтому внутренняя память большинства быстродействующих машин выполнена именно на ферритовых тороидах (машины БЭСМ, ЛАРК и др.).

Заметим, что применение ферритовых тороидов не ограничивается только устройствами памяти. Они с успехом применяются в логических и вычислительных схемах машин, причем с точки зрения надежности конкурируют с полупроводниковыми приборами.

Не менее интересным чем ферритовое устройство, является запоминающее устройство на электронно-лучевых трубках. Опишем кратко физические явления, с которыми тесно связана работа электронно-лучевой трубки памяти.

Известно, что если конденсатору сообщить электрический заряд, то спустя некоторое время этот заряд можно обнару-

жить. При очень хорошей изоляции между обкладками конденсатора заметный заряд сохраняется в течение многих суток. Вы уже догадались о том, что конденсатор можно использовать для запоминания двоичных цифр. Можно условиться, что заряженный конденсатор сохраняет изображение единицы, а незаряженный — нуль. Можно условиться и по другому, а именно: если одна из обкладок заряженного конденсатора несет на себе положительный потенциал, то конденсатор сохраняет изображение единицы, а если эта же обкладка находится под отрицательным потенциалом, то конденсатор сохраняет изображение нуля.

Нетрудно построить устройство для запоминания нескольких двоичных чисел, где в качестве запоминающих элементов будут служить конденсаторы. Однако при создании устройства по тому же принципу для запоминания хотя бы нескольких сотен чисел, встречаются большие трудности, которые не всегда удается преодолеть, если использовать конденсаторы в прямом смысле этого слова.

Второе физическое явление, о котором следует иметь представление, называется явлением вторичной эмиссии. Сущность вторичной эмиссии заключается в том, что при бомбардировке электронным потоком поверхности ряда веществ, из последних выбиваются электроны. Электроны потока называют первичными, а выбитые электроны вторичными. Количественная оценка вторичной эмиссии выражается в виде отношения числа вторичных электронов к числу первичных. Это отношение называется коэффициентом вторичной эмиссии. Коэффициент вторичной эмиссии для данного вещества сильно зависит от скорости первичных электронов. Если скорости первичных электронов небольшие, то коэффициент вторичной эмиссии меньше единицы. При увеличении скорости первичных электронов коэффициент вторичной эмиссии достигает значения единицы и далее становится больше единицы. Подчеркнем важное для нас обстоятельство, заключающееся в том, что при коэффициенте вторичной эмиссии больше единицы количество вторичных электронов больше, чем первичных.

Обратимся теперь к рис. 22, где дано схематическое изображение электронно-лучевой трубки памяти. Тот, кто знаком с электронно-лучевыми трубками телевизоров или осциллографов, заметит некоторое сходство между ними и изображенной трубкой памяти. Катод (*К*), подогреваемый нитью накала, выбрасывает электроны, которые фокусируются электронной пушкой в очень тонкий пучок или электронный луч (*ЭЛ*). Поток элек-

тронов от катода, а следовательно и луч, могут быть перекрыты, если на управляющую сетку подать соответствующее запирающее напряжение. С помощью напряжений на вертикальных и горизонтальных отклоняющих пластинах электронный луч направляется на любую, заранее намеченную элементарную площадку мишени (*М*). Число площадок на мишени определяется количеством цифр, которые предполагается запоминать трубкой. Мишень представляет собой тонкий слой диэлектрика, образованного на металлической сигнальной пластине (*СП*). Благодаря особой обработке мишень приобретает

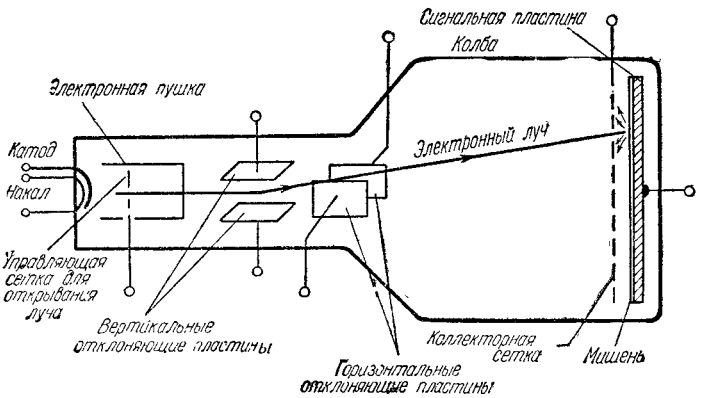


Рис. 22. Схема электронно-лучевой трубки.

свойство эмиттировать (испускать) вторичные электроны под действием первичных электронов луча. Скорость первичных электронов такова, что коэффициент вторичной эмиссии всегда больше единицы. Очень близко к мишени, над всей ее поверхностью, расположена частая коллекторная сетка (*КС*), предназначенная для улавливания вторичных электронов, которые вылетают с поверхности мишени в беспорядочных направлениях. Коллекторная сетка не оказывает препятствия быстрым электронам луча, но зато хорошо задерживает медленные вторичные электроны.

Опишем один из способов представления значений двоичных цифр посредством накопления электрических зарядов элементарными площадками мишени. Способ предполагает, что потенциалы катода и сигнальной пластины всегда остаются неизменными. Потенциал же коллекторной сетки при записи двоичных цифр изменяется относительно сигнальной пластины

в большую или меньшую сторону, т. е. может принимать некоторые положительные или отрицательные значения. Это достигается тем, что на вход коллекторной сетки подаются положительные или отрицательные импульсы напряжения в зависимости от того, какую цифру хотят записывать. Условимся, что для записи единицы, на коллекторную сетку подается положительный импульс напряжения, а для записи нуля — отрицательный.

На рис. 23 схематически изображены три различных состояния трубки, соответствующие записи единицы, нуля и чте-

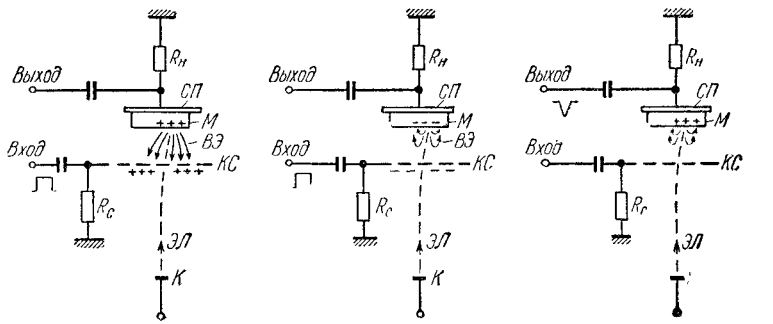


Рис. 23.

ния запомненной цифры, а также относительное распределение потенциалов на основных электродах трубки. Пусть электронный луч направляется отклоняющими пластинами на заданную площадку мишени и бомбардирует ее поверхность. Электроны луча, обладая большой скоростью, будут беспрепятственно пролетать коллекторную сетку и выбивать с поверхности диэлектрика вторичные электроны, количество которых намного больше, чем первичных.

Для записи единицы (левый рисунок) на вход коллекторной сетки подается положительный импульс напряжения. Разность потенциалов между коллекторной сеткой и сигнальной пластиной, которая находится под меньшим потенциалом, чем коллекторная сетка, приведет к тому, что подавляющая часть вторичных электронов будет притягиваться коллекторной сеткой. Теряя больше электронов, чем получая их, диэлектрик приобретет положительный заряд, а на его поверхности появится положительный потенциал, близкий к потенциалу коллекторной сетки. Процесс заряда диэлектрика легче

понять, если поток вторичных электронов представить себе в виде проводников, замыкающих поверхность диэлектрика с коллекторной сеткой. Не следует думать, что положительный потенциал получит вся поверхность диэлектрика. Действительно, заряд диэлектрика обусловлен исключительно потоком вторичных электронов, которые возникают только в месте падения электронного луча. Мы же условились, что в течение всего процесса записи электронный луч направлен в одно и то же место мишени. Кроме того, диэлектрические свойства мишени не позволяют заряду распространяться далеко за пределы элементарной площадки. В результате рассмотренного процесса, элементарный конденсатор, образованный коллекторной сеткой, диэлектриком и сигнальной пластиной, получит электрический заряд, который будет сохраняться достаточно долго даже тогда, когда потенциал коллекторной сетки изменится, а электронный луч будет заперт или направлен на другой участок мишени. Это состояние элементарного конденсатора и изображает единицу.

Для записи нуля (средний рисунок) на вход коллекторной сетки подается отрицательный импульс напряжения, а электронный луч направляется отклоняющими пластинами на избранный участок мишени. Отрицательный потенциал коллекторной сетки будет отталкивать вторичные электроны и возвращать подавляющую их часть на мишень. Диэлектрик будет накапливать отрицательный заряд в месте падения первичных и вторичных электронов, а поверхность элементарной площадки приобретает отрицательный потенциал, близкий к потенциалу коллекторной сетки. Заряд и потенциал, полученные при записи нуля, могут храниться так же долго, как и при записи единицы.

Считывание цифр из памяти осуществляется путем подачи электронного луча на избранную элементарную площадку мишени при потенциале коллекторной сетки, равном нулю. Как и в случаях записи единицы или нуля, элемент диэлектрика будет стремиться приобрести потенциал, близкий к потенциалу коллекторной сетки. Если при записи элемент был заряжен до положительного потенциала, то в первый момент времени вторичные электроны будут притягиваться поверхностью диэлектрика и в цепи сопротивления нагрузки $R_н$ возникнет импульс тока, совпадающий по направлению с движением вторичных электронов. Падение напряжения на сопротивлении $R_н$ вызовет появление отрицательного импульса напряжения на выходе трубки. Аналогично, рассматривая процесс чтения нуля, можно

заключить, что на выходе появится положительный импульс напряжения. Таким образом, по полярности импульса на выходе однозначно определяется считываемая цифра.

При считывании заряд элементарного конденсатора «портится», и чтобы информацию можно было считывать многократно, специальная схема регенерации восстанавливает заряд сразу же после чтения. В силу разных неблагоприятных причин заряды элементарных конденсаторов могут «портиться» с течением времени сами по себе и без всякого чтения, по-

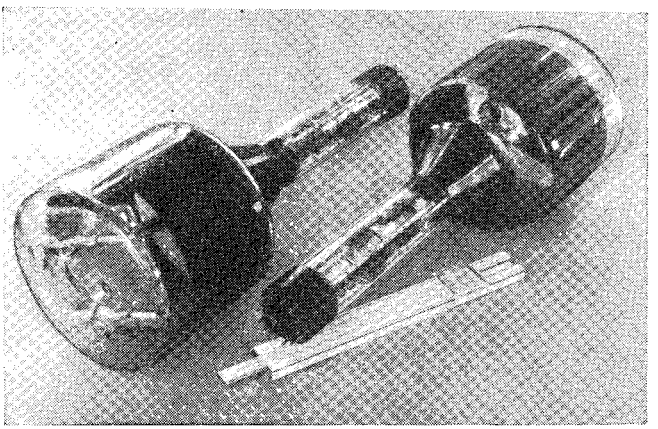


Рис. 24. Электронно-лучевые трубки памяти.

этому та же схема регенерации периодически их восстанавливает точно так же, как и при чтении.

Описанная электронно-лучевая трубка относится к так называемому типу потенциалоскопов и может запоминать до 4000 цифр. Время записи или чтения составляет примерно 4—6 *мксек*. Подобные трубки используются в машинах «Стрела» и БЭСМ.

На рис. 24 приведена фотография трубок. На переднем конце правой трубки видна электронная пушка, а левой трубки — сигнальная пластина.

На рис. 25 приведена фотография части стойки внутренней памяти машины «Стрела». Внизу видны восемь сменных блочков с трубками памяти. Выше расположены усилители и схемы регенерации.

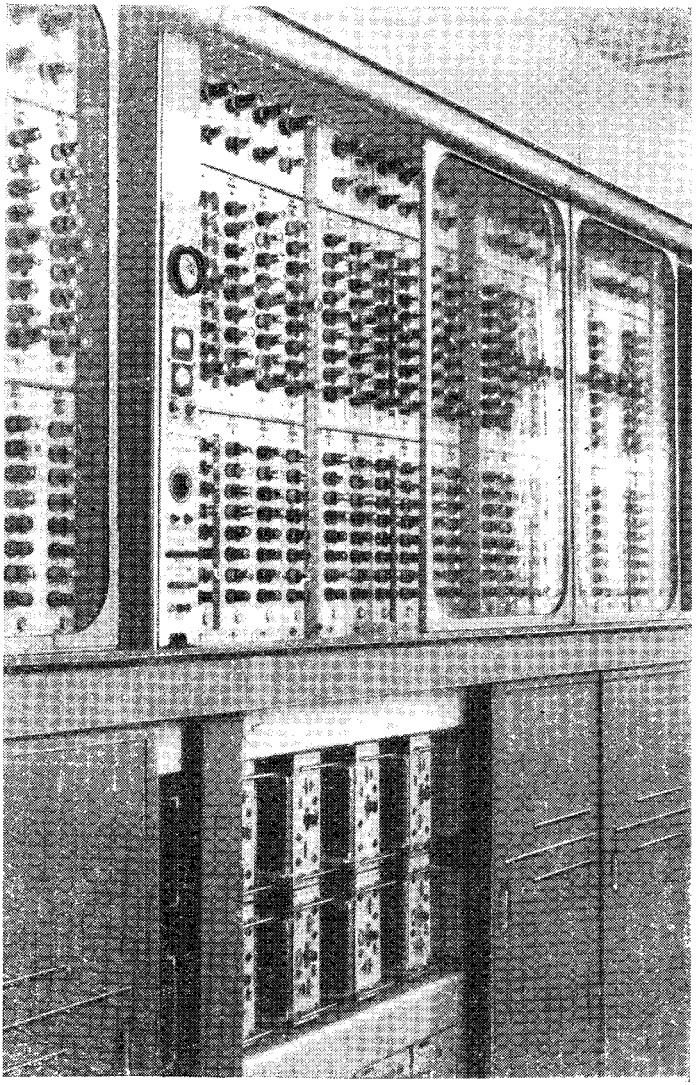


Рис. 25. Часть стойки внутренней памяти.

4. Устройства ввода и вывода

В предыдущих разделах мы уже кратко познакомились с устройствами ввода и вывода, или, как еще их называют, внешними устройствами, в которых используются перфокарты или перфоленты. Чтобы полнее представить состав комплекта внешних устройств, мы остановим внимание, например, на машине «Стрела».

Программа вычислений и исходные данные переносятся на перфокарты с помощью клавишного устройства (рис. 26) и входного перфоратора (рис. 27). Нажимая клавиши, оператор набирает очередное число или команду и стартовой кнопкой передает набор перфоратору.

На клавишном устройстве команды набираются в восьмеричной системе, а числа в десятичной системе счисления. Особая схема преобразует восьмеричный код в двоичный, а десятичный код — в двоично-десятичный.

Перфоратор пробивает каждый набор на перфокарте и передвигает ее в новую позицию. Чтобы исключить ошибки при перфорировании, подготавливаются два комплекта перфокарт разными операторами и на разных устройствах, а затем оба комплекта сличаются. Окончательно проверенный комплект перфокарт переносят в устройство ввода (рис. 28). Устройство ввода считывает информацию с перфокарт и передает ее в память машины. Результаты вычислений в двоично-десятичной системе передаются из запоминающего устройства машины на выходной перфоратор, конструкция которого мало чем отличается от входного перфоратора. Выходной перфоратор набивает результаты вычислений на перфокартах, которые затем переносят на печатающее устройство (рис. 29). Печатающее устройство подвигает перфокарты и «ощупывает» пробивки металлическими щеточками подобно тому, как это делает устройство ввода. Полученные электрические импульсы приводят в действие сложный печатающий механизм. Этот механизм переводит двоично-десятичный код в десятичный и печатает результаты на бумажной ленте в виде десятичных чисел.

Внешние устройства на перфокартах обладают сравнительно небольшим быстродействием. Так, ввод в машину осуществляется со скоростью около 20 чисел в сек., а вывод — со скоростью около 10 чисел в сек.

К более быстродействующим относятся устройства ввода и вывода с магнитной лентой, на которую записывается как

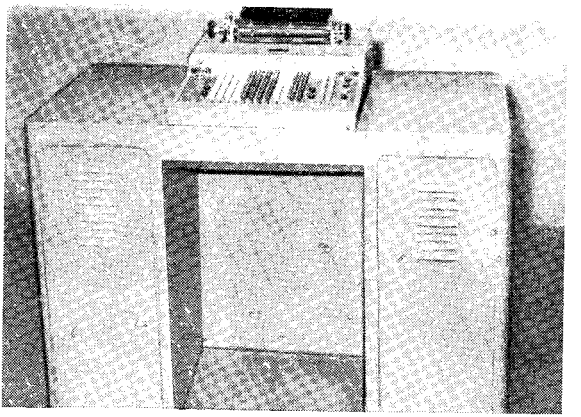


Рис. 26. Клавишное устройство.

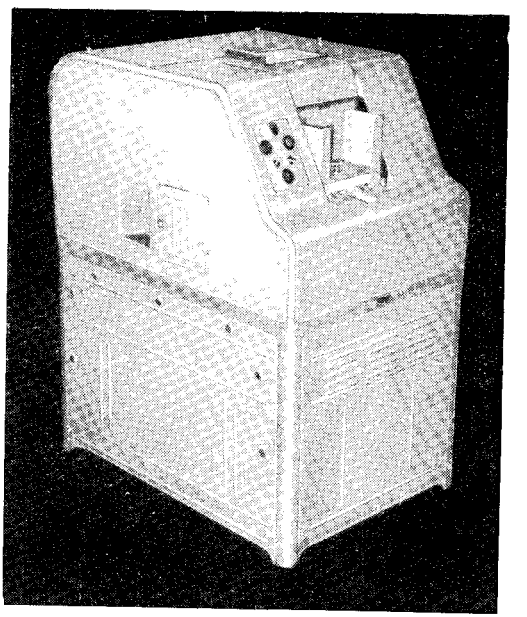


Рис. 27. Входной перфоратор.

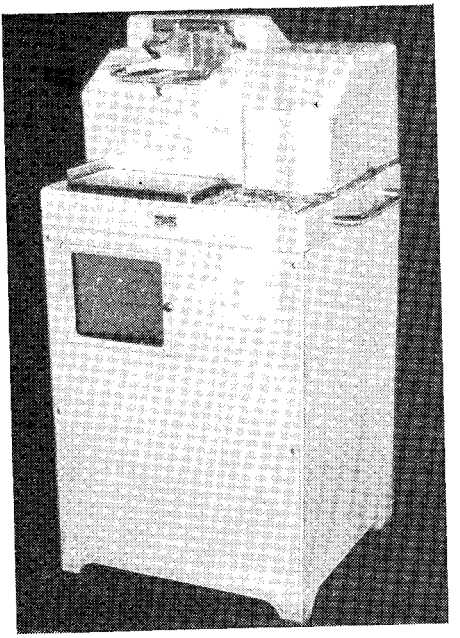


Рис. 28. Устройство ввода.

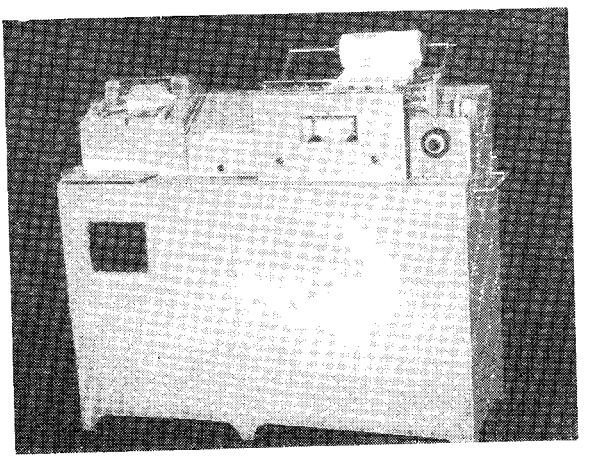


Рис. 29. Печатающее устройство.

программа вычислений, так и окончательные результаты. Скорость записи результатов или ввода программы с ленты достигает 300 чисел в сек. Некоторое распространение получили устройства с записью результатов на фотопленку сразу в виде десятичных чисел. Обладая огромной скоростью записи (около 2000 чисел в сек.), они тем не менее мало удобны, так как требуют громоздкого проявочного оборудования и не позволяют быстро знакомиться с результатами.

Недавно были открыты новые принципы считывания информации непосредственно с рукописи и передачи ее в зашифрованном виде в машину. Один из принципов состоит в том, что информация записывается на бумаге специальными чернилами, представляющими смесь красителя и тонко размолотого ферромагнитного порошка. Написанная такими чернилами рукопись с большой скоростью протягивается под магнитными головками. Каждая цифра (или буква) вызывает на выходе головки импульс напряжения вполне определенной формы, свойственной только данной цифре. Специальная схема распознает по форме импульса соответствующую ей цифру, зашифровывает ее на машинном языке и передает в запоминающее устройство.

Второй принцип состоит в том, что изображения цифр (или букв) проектируются на экран электронно-лучевой трубки, который представляет сетку из проволочек, натянутых друг к другу под прямым углом. Проволочки изолированы одна от другой. Световое изображение цифры покрывает большую часть проволочек экрана и вызывает на них появление электрических зарядов. Заряд и потенциал каждой проволочки примерно пропорциональны ее освещенности. Взаимное распределение зарядов и потенциалов на проволочках для каждой цифры вполне определенно, что и позволяет распознавать отдельные знаки и весь текст рукописи.

Предполагается, что использование подобных принципов даст возможность читать рукопись и вводить информацию в машину со скоростью нескольких тысяч знаков в секунду.

5. Контроль исправности

Как же быть уверенным в том, что вычислительная машина работает правильно, и как находятся и устраняются неисправности в такой сложной установке?

Один из известных способов контроля состоит в том, что на машине решаются специально подобранные задачи с зара-

нее известными ответами — эталонами. Задачи подбираются так, чтобы при их решении участвовали в работе все или подавляющее большинство деталей машины. Машина решает задачи испытательной программы, автоматически сравнивает полученные ответы с эталонами и в случае ошибки останавливается. При остановке, по горению сигнальных ламп на пульте, оператор может определить номер неправильно решенной задачи и повторить ее решение. Номер задачи и анализ решения подсказывают оператору, какое из устройств машины неисправно. Если все задачи испытательной программы решаются правильно, то можно быть уверенным в исправности машины. Однако эта

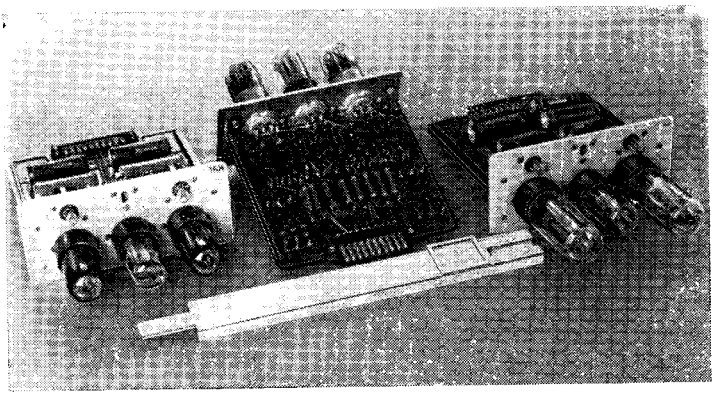


Рис. 30. Сменные ячейки.

так называемая текущая, или оперативная, проверка машины еще не дает гарантии ее исправности на несколько ближайших часов или суток. Такую гарантию дает профилактическая проверка.

Опыты показали, что детали машины редко выходят из строя катастрофически быстро. Например, такие наименее надежные элементы, как электронные лампы, чаще всего выходят из строя из-за потери эмиссии. Однако время, в течение которого лампа теряет эмиссию от еще приемлемого уровня до недопустимого, иногда тянется десятками и даже сотнями часов. Медленное изменение параметров ламп или других радиодеталей до некоторого времени может не сказаться на работе машины, и понятно, что было бы ценным заведомо заменять их на совершенно исправные. Но как обнаружить такие

детали? Для этой цели машину заставляют решать задачи испытательной программы в режиме измененных питающих напряжений. Значения напряжений выбираются такими, чтобы вполне исправные детали еще не вносили ошибок в работу машины. Зато детали с критическими или явно плохими параметрами обязательно внесут ошибки.

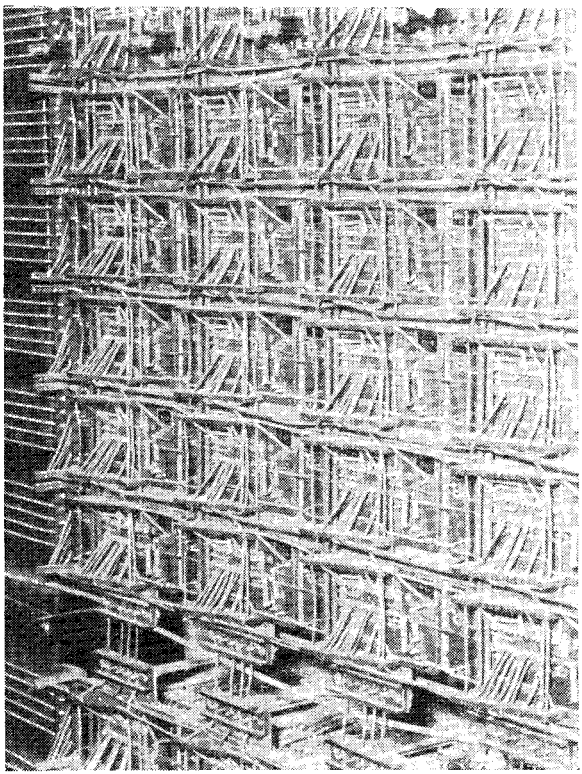


Рис. 31. Монтажная сторона стойки.

Теперь останется только найти и заменить ненадежные детали. В этом и заключается смысл профилактической проверки машины.

Большую гарантию от случайных ошибок машины дает «счет в две руки». Метод состоит в том, что одна и та же часть задачи решается два раза и результаты автоматически

IV. ПРОГРАММИРОВАНИЕ И РЕШЕНИЕ ЗАДАЧ

1. Команды

Нам известно, что запоминающее устройство вычислительной машины состоит из ячеек, в каждой из которых можно хранить только одно число. Каждой ячейке присвоен свой постоянный номер. В дальнейшем номера ячеек будем обозначать греческими или латинскими буквами. Часто, для того чтобы показать, что в ячейке с номером α хранится какая-то величина a, пишут $(\alpha)=a$ и говорят, что содержимое ячейки α равно a. Номер α ячейки памяти, содержащей величину a, называется адресом. Если число π имеет адрес 173, то $(173)=\pi$. Этим обозначением мы часто будем пользоваться.

Если известно, что $(\alpha)=a$ и $(\beta)=b$, то как получить числа $(\alpha)+(\beta)$, $(\alpha)-(\beta)$, $(\alpha)\cdot(\beta)$, $(\alpha):(\beta)$? Иными словами, как можно оперировать с числами, хранящимися в запоминающем устройстве ЭЦВМ и куда и как записывать их результаты?

Начнем со сложения. Пусть результат $(\alpha)+(\beta)$ мы хотим записать в ячейку с номером γ. Словесная формулировка такой задачи выглядит так: «сложить содержимое ячейки с номером α с содержимым ячейки с номером β и результат этого действия записать в ячейку с номером γ», или символически:

$$+\,\alpha\,\beta\,\gamma.$$

Такая строчка, в которой указывается вид операции (сложение, вычитание и т. д.), объекты этой операции (два числа, хранящиеся в ячейках α и β, и место записи результата (ячейка γ), образует *команду*. Команда в зашифрованном (обычно двоичными цифрами) виде поступает в одно из устройств машины, где она расшифровывается машиной и выполняется. Подробнее о том, в какое именно устройство и в каком виде поступает команда, мы расскажем ниже.

Итак, после выполнения команды сложения будем иметь $(\alpha) + (\beta) = (\gamma)$. Наша команда содержит три адреса величин, участвующих в сложении. Мы будем нумеровать эти адреса по порядку слева направо: 1-й адрес, 2-й адрес, 3-й адрес. Условимся в дальнейшем знак операции писать в конце команды:

1-й адрес	2-й адрес	3-й адрес	Вид операции
α	β	γ	+

Подчеркнем еще раз, что α означает лишь номер ячейки, где хранится одно число, а β — номер ячейки, где хранится другое число. Иными словами, α и β есть лишь номера (адреса), по которым заранее ничего нельзя сказать о величине хранимых в них чисел, подобно тому как номер дома еще ничего не говорит об его обитателях. Это замечание следует помнить и впредь.

По образцу команды сложения можно записать и команду вычитания $(\alpha) - (\beta)$:

α	β	γ	−

после выполнения которой в нашем случае $(\alpha) - (\beta) = (\gamma)$.

Команда умножения $(\alpha) \cdot (\beta)$ имеет вид:

α	β	γ	×

и для нашего случая $(\alpha) \cdot (\beta) = (\gamma)$.

Команда деления $(\alpha) : (\beta)$ запишется так:

α	β	γ	:

и $(\alpha) : (\beta) = (\gamma)$.

Последняя команда имеет смысл, когда $(\beta) \neq 0$, т. е. содержимое ячейки, записанной по второму адресу в команде деления, должно быть отлично от нуля. В противном случае происходит автоматическая остановка работы машины.

Кроме этих четырех команд, с помощью которых можно производить арифметические действия, существует еще целый ряд других, которые образуют систему команд для машин данного типа. Приведем примеры двух «неарифметических» команд, нужных нам в дальнейшем.

Иногда нужно переписать все содержимое ячейки α в ячейку β, при этом, конечно, все предыдущее содержимое ячейки β стирается. Для этого применяется операция засылки (обозначим «$\longrightarrow$»). Соответствующая команда будет:

α	0	β	$\longrightarrow$

В этой операции участвуют лишь две ячейки, поэтому во втором адресе стоит нуль.

С другими операциями и командами мы познакомимся в следующем разделе, посвященном программированию. Здесь же отметим, что количество различных команд, т. е. система команд, зависит от количества различных операций, которые может совершать данная машина. Последнее в конечном счете зависит от конструктивных особенностей ее. Поэтому обычно, прежде чем конструировать машину, конструктор должен знать хотя бы примерный круг задач, которые будут решаться на ней: будет ли это специальная машина, машина универсальная и т. п.

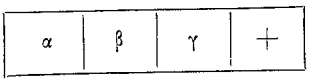
Чем разнообразнее операции данной машины, чем шире их круг, тем проще решать задачи на такой машине. Но с другой стороны, это приводит и к конструктивному усложнению всей машины. Отсюда возникает задача выработки так называемой оптимальной системы команд, при которой машина могла бы делать как можно больше различных операций и имела бы не слишком сложную структуру.

Команды, с которыми мы только что познакомились, содержат три адреса. Это — трехадресная система команд. Такие вычислительные машины, кам БЭСМ, «Стрела» являются трехадресными. Кроме машин с трехадресной системой команд, существуют одноадресные, двухадресные и четырехадресные машины. Коротко скажем о каждой из таких машин.

В одноадресных машинах существует, кроме известной нам памяти, еще одно быстродействующее запоминающее устройство на одно число, которое расположено в арифметическом устройстве.

Это устройство называется регистром. Регистр нужен для того, чтобы хранить промежуточные результаты при арифметических операциях.

Операция сложения в трехадресной системе команд:



распадается на следующие три одноадресные команды:

1. $+\alpha$
2. $+\beta$
3. $\rightarrow\gamma$.

Первая команда сложит содержимое регистра (предполагая, что там стоит нуль) с содержимым ячейки α и запишет результат в регистр. Вторая сложит содержимое регистра (число (α) с числом (β) и запишет результат в регистр. Последняя команда зашлет содержимое регистра в ячейку γ.

В конструктивном отношении одноадресные машины значительно проще трехадресных. Поэтому в некоторых как проектируемых, так и работающих машинах применяется одноадресная система команд (у нас в СССР «Урал», в Англии ЭДСАК, в США ИБМ-701).

В двухадресной команде не указывается номер ячейки результата, так как последний записывается в ячейку одного из исходных чисел, которое при этом стирается. Двухадресная система команд принята, например, на машине МАРК-II (США).

В четырехадресной команде добавляется еще один адрес, в котором указывается, к какой команде следует перейти после выполнения первой. Это — так называемый принудительный порядок выполнения команд. Примером четырехадресной машины может служить Райтеон (США).

Находят применение в настоящее время и «полутораадресные» машины, но на них мы останавливаться не будем.

Естественно спросить: какая из машин лучше? С точки зрения математика возможности машины зависят не от количества адресов в их командах. Поэтому в вычислительном отношении все эти машины примерно одинаковы. Решение вопроса о том, какую предпочесть машину, одноадресную или трехадресную, зависит скорее от конструктора, чем от перспектив ее математического использования.

2. Программирование задач для цифровых вычислительных машин

Мы познакомились с некоторыми арифметическими и неарифметическими операциями, совершаемыми машиной, и с командами, нужными для выполнения этих операций.

Как же применяются эти команды при решении задач? Начнем с простого примера.

Пример 1. Пусть нужно рассчитать путь s равномерно-ускоренного движения за какое-то время t. Из физики известно, что эта зависимость выражается следующей формулой:

$$s = v_0 t + \frac{at^2}{2},$$

где v_0 — начальная скорость и a — ускорение, причем v_0 и a — постоянные для этого движения величины.

Последовательность операций, нужных для нахождения s, можно было бы описать и без формулы, например, так: «Результат возведения t в квадрат умножить на a и полученное произведение умножить на $\frac{1}{2}$. Полученную величину $\frac{at^2}{2}$ сложить с результатом умножения v_0 на t». Разумеется, что никакой словесной формулировки машина понять не может, для этого нужен свой, понятный для машины «язык». Оказывается, что такой «язык» действительно существует. Попробуем перевести нашу задачу на этот машинный «язык». Предположим, что исходные величины уже размещены в следующих ячейках запоминающего устройства:

$$(\alpha) = v_0,\ (\beta) = a,\ (\gamma) = \frac{1}{2},\ (\delta) = t.$$

Первое предложение словесной формулировки нашей задачи в виде, «понятном» для машины, может быть записано как три команды умножения:

$$1.\ \times\ \delta\ \ \delta\ \ r_1, \quad \text{т. е. } (r_1) = t^2.$$

Теперь нужно t^2 умножить на a, или на «языке» машины:

$$2.\ \times\ r_1\ \ \beta\ \ r_2, \quad \text{т. е. } (r_2) = at^2$$

и

$$3.\ \times\ r_2\ \ \gamma\ \ r_3, \quad \text{т. е. } (r_3) = \frac{at^2}{2}.$$

Второе предложение можно записать так:

$$4.\ \times\ \alpha\ \ \delta\ \ r_4, \quad \text{т. е. } (r_4) = v_0 t.$$

$$5.\ +\ r_4\ \ r_3\ \ r_5, \quad \text{т. е. } (r_5) = s.$$

Задача вся переведена на машинный язык. Нужно только еще предусмотреть команду остановки, которую мы пока обозначим «стоп», т. е.

6. Стоп

Анализируя занятые теперь нами новые ячейки r_1, r_2, r_3, r_4, r_5, где хранятся промежуточные результаты, видим, что не все эти результаты нужно хранить до конца. Например, после того как найдена величина at^2, величина t^2 нам не нужна; не нужна и v_0t после того, как мы сложили ее с $\frac{at^2}{2}$. Поэтому вместо, например, ячейки r_2 можно употребить ячейку r_1, поставив туда результат умножения a на t^2. Напомним, что машина сконструирована так, что, какое бы число ни стояло до этого в ячейке, после постановки туда нового числа первое будет стерто.

Перепишем теперь всю последовательность операций, необходимых для решения нашей задачи, учитывая замечание об ячейках с промежуточными результатами:

1.	δ	δ	r_1	$\times$	$(r_1) = t^2$,
2.	r_1	β	r_1	$\times$	$(r_1) = at^2$,
3.	r_1	γ	r_1	$\times$	$(r_1) = \frac{at^2}{2}$,
4.	α	δ	r_2	$\times$	$(r_2) = v_0t$,
5.	r_1	r_2	r_2	$+$	$(r_2) = s$.
6.				стоп	

Такая последовательность команд образует *программу* решения нашей задачи на машине. Она вводится в машину в специально закодированном виде, о чем подробно мы скажем ниже, и ее команды автоматически выполняются машиной одна за другой. Ясно, что переставлять команды в уже готовой программе, вообще говоря, нельзя, так как это нарушит всю структуру выбранной последовательности операций и нумерацию ячеек.

Составленная нами программа позволяет, очевидно, вычислять s при любых t, стоит лишь заслать в ячейку, где хранится t, новую величину и повторить вычисления снова.

Для громадного большинства задач, решаемых на быстродействующих машинах, характерным является то обстоятельство, что вычислительный процесс может как бы «раздва-

иваться», т. е. существуют условия, при выполнении которых нужно, скажем, вести счет по одной формуле, а при невыполнении этих условий — по другой. Для полной автоматизации счета необходимо, чтобы машина, когда это нужно, проверяла сама эти условия и вела бы счет то по одной формуле, то по другой, запрограммированных заранее программистом. С помощью только арифметических операций этого сделать не удастся: нужны совершенно иные по своему характеру и назначению операции, которые называются логическими.

Как же осуществить подобное раздвоение счета в программе? При выполнении некоторых операций машина наряду с результатом операции вырабатывает особый сигнал, который может быть либо 0, либо 1. Этот сигнал не предусматривается программой, машина вырабатывает его сама и хранит до тех пор, пока не будет выполнена команда, следующая за той, по которой машина выработала этот сигнал, после чего он гасится или вырабатывается вновь. Этот сигнал в машине «Стрела» вырабатывается, например, при выполнении команды вычитания; он равен нулю, если результат положителен или нуль, и единице, если результат отрицателен. Если обозначить этот сигнал буквой ω, то при вычитании имеем:

$$\text{если } z = x - y, \text{ то } \omega = \begin{cases} 1 \text{ при } z < 0, \\ 0 \text{ при } z \geqslant 0. \end{cases}$$

Совершенно аналогично он вырабатывается и при команде сложения:

$$\text{если } z = x + y, \text{ то } \omega = \begin{cases} 1 \text{ при } z < 0, \\ 0 \text{ при } z \geqslant 0. \end{cases}$$

Часто бывает необходимо сравнить два числа (α) и (β), т. е. узнать, равны они или нет. Для этого применяется команда «сравнение». Ее мы обозначим условно как «Ср».

Сравнение чисел (α) и (β) происходит по всем разрядам разрядной сетки. Эта команда имеет вид:

α	β	0	Ср

При этом $\omega = 0$, если $(\alpha) = (\beta)$ и $\omega = 1$, если $(\alpha) \neq (\beta)$.

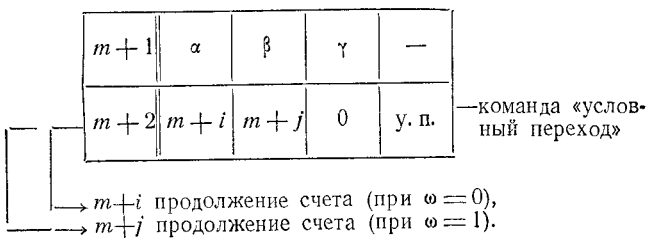
Однако сам по себе сигнал ω еще не позволяет «ветвить» вычислительный процесс. Для этого существует особая операция, называемая «условный переход» (коротко «у. п.»). Раз-

берем, как работает операция «у. п.». Пусть в программе стоит команда

$m+1$	α	β	γ	—

где $m+1$ — порядковый номер этой команды в программе.

Тогда при выполнении этой команды вырабатывается сигнал ω, равный нулю или единице. Пусть известно, что если $\omega=0$, то нужно перейти к выполнению команды, стоящей на $(m+i)$ месте, а при $\omega=1$ — к команде, стоящей на $(m+j)$ месте. Это и осуществляет операция «у. п.»: по 1-му адресу этой команды ставится порядковый номер той команды, с которой нужно продолжить счет при $\omega=0$, а по 2-му адресу — номер команды для продолжения счета при $\omega=1$ (по 3-му адресу будем записывать нуль). Это можно изобразить такой схемой:



Посмотрим, как работает операция «у. п.» в целой программе.

Пример 2. Составим программу решения такой задачи: найти сумму S_n последовательных натуральных чисел, т. е.

$$S_n=1+2+3+\cdots+n \quad (n\text{ — задано}).$$

Основным соотношением для нас будет следующая зависимость:

$$S_{i+1}=S_i+(i+1);\ i=0,\ 1,\ 2,\ \ldots,\ n-1;\ S_0=0.$$

Для решения такой задачи нам потребуется число 1 и число $n-1$ (последнее значение i). Пусть эти два числа хранятся в ячейках: $1=(c_1)$; $n-1=(c_2)$.

Тогда программа для любого заданного натурального n будет:

	$m+1$	r_1	c_1	r_1	$+$	в начале $(r_1)=0$,
$\omega=0$	$m+2$	α	r_1	α	$+$	в начале $(\alpha)=0$,
	$m+3$	c_2	r_1	r_2	$-$	
	$m+4$	$m+1$	$m+5$	0	у. п.	
$\omega=1$	$m+5$				стоп	результат стоит в α.

Поясним ее. Команда $m+1$ образует сумму $i+1$, команда $m+2$ складывает предыдущую сумму S_i с $i+1$. Команда $m+3$ нужна для выработки сигнала ω, чтобы закончить счет при заданном n. Команда $m+4$ в зависимости от выработанного значения ω передает управление либо команде $m+1$ (при $\omega=0$, когда $(c_2)\geqslant(r_1)$), либо команде $m+5$ ($\omega=1$, когда $(c_2)<(r_1)$).

Читателю рекомендуется, задавшись каким-либо n, проследить по этой программе всю последовательность операций, записывая промежуточные результаты на листке бумаги.

Отметим важную особенность этой программы: ее «длина» не зависит от n, хотя, разумеется, время счета будет существенно зависеть от него. Это объясняется тем, что часть программы многократно повторяется. Такое повторение называется *циклом*. В нашей программе — это последовательность команд от $m+1$ до $m+4$.

Наличие таких циклов, а их может быть и несколько в одной и той же программе, является еще одной характерной особенностью многих задач. Циклическая программа имеет очевидное преимущество перед последовательным расписыванием всех команд, число которых в этом случае существенно зависит от исходных данных.

Если задано n, то нетрудно подсчитать, сколько раз будет пройдена последовательность команд от $m+1$ до $m+4$ (сколько будет циклов). Но существуют задачи с циклическим счетом, где этот подсчет сделать нельзя. Это — задачи с итерационным методом счета. К ним мы сейчас и перейдем.

Известно правило, по которому извлекается квадратный корень из числа. Эту последовательность операций, определенную на каждом этапе счета (такого рода правила часто называют алгорифмом *)), можно было бы запрограммировать. Но гораздо удобнее с точки зрения программирования и счета является построение такой формулы и алгорифма, чтобы при многократном повторении счета по ней мы могли бы получить значение корня с какой угодно малой погрешностью, т. е. «подойти» к истинному значению как угодно близко. Счет по таким формулам называется итерацией от латинского слова iteratio — повторение.

Мы сейчас выведем такую формулу для извлечения квадратного корня, которая может быть полезна и при ручном счете.

Пусть надо вычислить

$$y=\sqrt{x}.$$

Будем считать, что мы сумели как-то определить некоторое, пусть весьма грубое, приближенное значение этого корня x_0, которое разнится от истинного y на величину h, где h мало по сравнению с x_0. Тогда

$$y=x_0+h$$

или

$$x=y^2=(x_0+h)^2=x_0^2+2x_0h+h^2.$$

Так как h относительно мало, то h^2 и подавно мало, и следовательно, можно приближенно написать:

$$x\approx x_0^2+2x_0h,$$

откуда

$$\frac{x-x_0^2}{2x_0}\approx h.$$

Таким образом,

$$y\approx x_0+\frac{x-x_0^2}{2x_0}=\frac{x+x_0^2}{2x_0},$$

или

$$y\approx\frac{1}{2}\left(x_0+\frac{x}{x_0}\right).$$

---

*) Алгорифм — всякая система операций, выполняемых по строго определенным правилам, которая после конечного числа шагов заведомо приводит к решению поставленной задачи.

Это и есть искомая формула. Здесь x — заданное число, в качестве же x_0 (первого приближения) можно взять, например, $\frac{x}{2}$. Удобно в нашей формуле несколько переменить обозначения: найденное первое приближение корня обозначим через x_1, т. е.

$$x_1 = \frac{1}{2}\left(x_0 + \frac{x}{x_0}\right).$$

Зная x_1, можем найти второе приближение, которое найдем по той же формуле, т. е.

$$x_2 = \frac{1}{2}\left(x_1 + \frac{x}{x_1}\right).$$

Вообще, зная приближение на n-м шаге, найдем $(n+1)$-е приближение:

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{x}{x_n}\right),$$

где $x_0 = \frac{x}{2}$.

В таком виде этой формулой обычно и пользуются.

Можно показать (здесь мы не будем на этом останавливаться), что формула $x_{n+1} = \frac{1}{2}\left(x_n + \frac{x}{x_n}\right)$ позволяет сколь угодно близко «подойти» к точному значению $\sqrt{x}$.

Проиллюстрируем счет по этой формуле, найдя $\sqrt{2}$. Здесь $x_0 = 1$:

$$x_1 = \frac{1}{2}\left(1 + \frac{2}{1}\right) = 1{,}5,$$

$$x_2 = \frac{1}{2}\left(1{,}5 + \frac{2}{1{,}5}\right) = 1{,}416\,667,$$

$$x_3 = \frac{1}{2}\left(1{,}416\ldots + \frac{2}{1{,}416\ldots}\right) = 1{,}414\,216,$$

$$x_4 = \frac{1}{2}\left(1{,}414\ldots + \frac{2}{1{,}414\ldots}\right) = 1{,}414\,214.$$

Уже на 4-м шаге мы получили значение $\sqrt{2}$ с точностью до 10^{-5}.

Как видно из этого примера, мы последовательно приближаемся к истинному значению корня, которое, конечно, не может быть найдено точно, ибо $\sqrt{2}$ — число иррациональное.

Возникает вопрос: на каком шаге нужно остановиться? Последовательные приближения $x_0, x_1, x_2, \ldots, x_n, \ldots$ по мере увеличения n все меньше и меньше отличаются друг от друга

$$\begin{aligned} |x_1 - x_0| &= 0{,}5 \\ |x_2 - x_1| &= 0{,}083\,333, \\ |x_3 - x_2| &= 0{,}002\,451, \\ |x_4 - x_3| &= 0{,}000\,002, \\ &\ldots\ldots\ldots\ldots \end{aligned}$$

Если теперь нам заранее известна та точность, с которой должен быть вычислен корень (эту заданную наперед точность обозначим через ε), то наше вычисление нужно закончить, когда будет

$$|x_{n+1} - x_n| < \varepsilon,$$

причем, разумеется, нам неизвестно число шагов n, когда будет соблюдено это условие.

Пример 3. Составим программу нахождения квадратного корня из заданного числа x. Пусть $(a) = x$; $(\delta_1) = \frac{1}{2}$, $(\delta_2) = \varepsilon$.

Последовательность команд для нахождения первого приближения будет:

$m+1$	a	δ_1	r_1	$\times$
$m+2$	r_1	0	r_2	$\longrightarrow$
$m+3$	a	r_2	r_1	:
$m+4$	r_2	r_1	r_1	$+$
$m+5$	δ_1	r_1	r_1	$\times$

$$(r_1) = x_0 = \frac{x}{2},$$
$$(r_2) = (r_1) = \frac{x}{2},$$
$$(r_1) = \frac{x}{x_0},$$
$$(r_1) = x_0 + \frac{x}{x_0}.$$
$$(r_1) = \frac{1}{2}\left(x_0 + \frac{x}{x_0}\right) = x_1.$$

Теперь весь этот цикл нужно повторить сначала, имея уже x_1 вместо x_0. Предварительно нужно проверить, не дошли ли мы до конца счета, т. е. найти $|x_1 - x_0| - \varepsilon$.

Если эта разность есть положительное число, то нужно повторить весь цикл сначала, начиная с команды $m+2$, которая поставит x_1 вместо x_0; если эта разность отрицательна, то нужно закончить вычисление, так как необходимая точность достигнута.

Операцию нахождения $|x_{n+1} - x_n|$ обозначим «$|-|$». Окончательно программа примет вид:

	$m+1$	a	δ_1	r_1	$\times$
	$m+2$	r_1	0	r_2	$\rightarrow$
	$m+3$	a	r_2	r_1	:
	$m+4$	r_2	r_1	r_1	$+$
	$m+5$	δ_1	r_1	r_1	$\times$
	$m+6$	r_1	r_2	r_3	$\|-\|$
	$m+7$	r_3	δ_2	0	$-$
$\|x_{i+1} - x_i\| - \varepsilon > 0 \quad (\omega = 0)$	$m+8$	$m+2$	$m+9$	0	у. п.
	$m+9$				стоп

$(r_3) = |x_1 - x_0|$ — первый раз,

результат записан в ячейку r_1.

Поясним команды $m+7$ и $m+8$. Так как нам нужен лишь знак результата вычитания в команде $m+7$, от чего будет зависеть значение ω, но не сам результат, то по 3-му адресу можно писать нуль (в этом случае результат никуда не запишется). Команда «у. п.» воспримет сигнал ω и передаст управление либо команде $m+2$, либо команде $m+9$ (стоп).

В этой программе заранее уже ничего нельзя сказать о числе пройденных циклов от $m+2$ до $m+8$, так как это будет зависеть как от заданного ε, так и от x. Но этого нам здесь знать и не нужно: построенная программа закончит вычисление лишь тогда, когда будет соблюдена заданная точность.

Методом итераций часто пользуются, особенно в алгебре для решения систем линейных алгебраических уравнений. Метод итераций обладает важной особенностью: он обеспечивает самоисправление ошибок: если в процессе итерации была допущена случайная относительно небольшая ошибка, то она не повлияет на точность результата, а разве лишь приведет к увеличению числа итераций, необходимых для получения нужной точности.

Как мы только что убедились, логическая операция «у. п.» позволяет менять порядок счета в зависимости от сигнала ω предыдущей команды. Видоизменяя команду «у. п.», можно получить новые команды, посредством которых выполняются другие логические операции. Мы рассмотрим здесь две из них.

Часто бывает необходимо нарушить последовательность выполняемых машиной команд вне зависимости от сигнала ω предыдущей команды. Для этого существует команда «безусловного перехода». Она получается из команды «у. п.», если по 1-му и 2-му адресу записать порядковый номер команды, которой следует передать управление:

$m+j$	. . .	. . .	. .	. .
$m+j+1$	$m+k$	$m+k$	0	у. п.

— команда безусловного перехода к $(m+k)$-й команде.

Здесь уже операция «у. п.» воспринимается как «безусловная» (независимо от ω) передача управления команде $m+k$.

Как мы отмечали, при выполнении команды «у. п.» по 3-му адресу машина записывает нуль, т. е. если в этом адресе будет записан номер какой-либо команды, то после выполнения операции «у. п.» в памяти машины вместо нее запишется нуль, команда сотрется в памяти. Этим можно воспользоваться, чтобы получить «самогасящую» команду. Она получается из «у. п.», если по 3-му адресу записать номер ее самой, т. е.

$m+j+1$	$m+k$	$m+k$	$m+j+1$	у. п.

и на $(m+j+1)$-м месте после выполнения ее будет стоять нуль.

Кроме «у. п.» и ее модификаций, другой важной операцией является «специальное сложение» (обозначим ее $\oplus$) и операция «специального вычитания» ($\ominus$). Мы разберем на примере операцию «специального сложения».

Пример 4. Пусть требуется перемножить пять чисел, записанных в подряд стоящих ячейках $\alpha+1$, $\alpha+2$, $\alpha+3$, $\alpha+4$, $\alpha+5$ на число, стоящее в ячейке γ, и результаты поставить опять-таки в подряд стоящие ячейки $\beta+1$, $\beta+2$,

$\beta+3$, $\beta+4$, $\beta+5$. Непосредственное программирование такой задачи выполнить просто. Вот эта программа:

$m+1$	$\alpha+1$	γ	$\beta+1$	$\times$
$m+2$	$\alpha+2$	γ	$\beta+2$	$\times$
$m+3$	$\alpha+3$	γ	$\beta+3$	$\times$
$m+4$	$\alpha+4$	γ	$\beta+4$	$\times$
$m+5$	$\alpha+5$	γ	$\beta+5$	$\times$
$m+6$				стоп

Присмотримся к ней повнимательнее: каждая команда, начиная со второй, отличается от предыдущей только тем, что 1-й адрес и 3-й адрес увеличиваются на единицу. Если бы нам удалось каким-либо образом проделывать это каждый раз с помощью самой машины, то из первой команды мы получили бы все остальные. Для этой цели заготовим такую константу:

000 000 000 001 000 000 000 000 000 000 001 0 00

или в восьмеричной записи:

0001 0000 0001 000,

а затем «прибавим» ее специальным образом к первой команде так, чтобы изменились на единицу 1-й и 3-й адреса, а код операции и 2-й адрес остались те же (для чего на этих местах в константе стоят нули). Операция «специального сложения» позволяет сделать это. Пусть эта константа записана в ячейке c_1. После того, как выполнена команда $m+1$, следует предусмотреть такую команду:

	$m+1$	c_1	$m+1$	$\oplus$

и после ее выполнения вместо прежней команды на $(m+1)$-м месте будет стоять команда:

$m+1$	$\alpha+2$	γ	$\beta+2$	$\times$

Теперь следует выполнить эту команду и снова ее переадресовать и т. д.

Но у нас всего пять чисел, поэтому эту переадресацию надо «вовремя» остановить. Это можно сделать, например, так: заготовим заранее еще такую константу-команду:

$(c_2) =$

$\alpha + 6$	γ	$\beta + 6$	$\times$

смысл которой будет ясен из дальнейшего, и предусмотрим такие команды:

$m+1$	$\alpha+1$	γ	$\beta+1$	$\times$
$m+2$	$m+1$	c_1	$m+1$	$\oplus$
$m+3$	$m+1$	c_2	0	ср.
$m+4$	$m+5$	$m+1$	0	у. п.
$m+5$				стоп

После выполнения первой команды произойдет ее переадресация и сравнение с константой c_2 (они не будут совпадать). При этом операция сравнения выработает признак $\omega = 1$ («не равны») и операция «у. п.» отошлет нас снова к команде $m+1$, где к этому времени будет уже стоять

$m+1$	$\alpha+2$	γ	$\beta+2$	$\times$

Эта команда выполнится, вновь переадресуется и снова произойдет сравнение. Последний раз после переадресации $(m+1)$-я команда будет

$m+1$	$\alpha+6$	γ	$\beta+6$	$\times$

Ее содержание лишено смысла для нашей задачи, но она и не будет выполнена, так как операция сравнения обнаружит совпадение ее с константой, выработает признак $\omega = 0$, операция «у. п.» при этом отошлет нас на «стоп» и цикл закончится. Задача решена.

Только что рассмотренная операция $\oplus$ является для нас совершенно новой, ибо здесь мы оперируем не с содержимым ячеек, а с их номерами. Однако здесь нет ничего нового для

самой машины. В самом деле, в ячейке, скажем $m+1$, записано:

$$\alpha+1 \quad \gamma \quad \beta+1 \quad \times.$$

Машине «безразлично», стоит ли в этой ячейке число или команда: ведь и то, и другое есть лишь набор единиц и нулей. Поэтому изменение адресов по операции «специальное сложение» машина произведет путем прибавления единиц в последние разряды этих модифицируемых адресов *).

Вернемся к нашей последней программе. Ее легко распространить на случай n чисел, стоящих подряд, для чего нужно иметь следующую константу для сравнения:

$$(c_2)=\boxed{\alpha+n+1 \mid \gamma \mid \beta+n+1 \mid \times}$$

и применить снова ту же программу. И здесь ее «длина» не зависит от n.

Очевидны удобства операции «специального сложения»: она позволяет значительно сократить написание программы, делает ее более универсальной.

Заканчивая на этом наше краткое знакомство с основами программирования, отметим, что составление программ для сложных задач требует большого и кропотливого труда со стороны программиста-математика, так как даже небольшая описка в написании числа или команды приводит иногда к негодности всей программы. Отсюда понятно стремление автоматизировать программирование. Такая автоматизация достигается тем, что программа составляется не «вручную», а с помощью самой машины, в которую для этого вводится особая программа, которая называется программирующей. Последняя является очень сложной по своей структуре и большой по объему, она позволяет обслуживать широкий класс задач при составлении для них программ на машине.

Существенный вклад как в теорию программирования, так и в теорию автоматизации программирования сделали советские математики.

---

*) Впервые мысль о возможности модификации команд самой машиной была высказана американским математиком Д. Нейманом в 1947 г.

Упражнение V

1) Составьте программу решения неоднородной системы алгебраических уравнений:

$$\left.\begin{aligned} a_1x + b_1y = c_1, \\ a_2x + b_2y = c_2 \end{aligned}\right\}$$

с коэффициентами, отличными от нуля. Предусмотрите в программе выделение случая несовместности (определитель системы равен нулю). Считать число нуль записанным в ячейке с тем же номером, т. е. $(0) = 0$.

2) Составьте программу нахождения действительных корней квадратного уравнения:

$$ax^2 + bx + c = 0,$$

где a, b, c — заданные числа, отличные от нуля. Предусмотрите возможность случая, когда $b^2 - 4ac < 0$ (комплексные корни). Операцию извлечения квадратного корня считать заданной в машине.

3) В некоторых машинах (например, «Стрела») операция деления как таковая отсутствует. Деление же одного числа на другое получается путем нахождения величины, обратной делителю. Нахождение величины «$\frac{1}{x}$» происходит по стандартной программе, постоянно хранящейся в машине.

Составьте программу нахождения «$\frac{1}{x}$» по итерационной формуле:

$$x_{n+1} = x_n(2 - x \cdot x_n),$$

где x_0 — любое число, удовлетворяющее неравенству

$$|1 - x \cdot x_0| < 1.$$

В программе предусмотрите выбор x_0, удовлетворяющего этому неравенству. Точность вычисления — заданное число ε.

3. Подготовка задачи к программированию и решение ее на цифровой вычислительной машине

Проследим весь тот путь, который проходит программа, начиная с ее составления и кончая выводом результатов счета из машины.

Прежде чем начать составлять программу, математик должен выбрать метод решения задачи. Часто таких методов бывает несколько. Математик выбирает тот из них, который быстрее приводит к решению, дает более точный результат, проще программируется.

Для выбора метода решения часто требуются серьезный математический анализ и высокая квалификация математика. После этого приступают непосредственно к программированию.

Если задача велика, то ее расчленяют на куски, программируют их, а затем «склеивают» в одну единую программу. При программировании следует всегда иметь в виду ту машину, на которой будет решаться данная задача, так как программа, составленная для одной машины, не может быть введена для решения в машину другого типа, на которой может быть принята совершенно иная система команд.

Как мы видели, в командах программы пишутся номера ячеек памяти. Они нумеруются четырехзначными числами в восьмеричной системе. Так, в машинах типа «Стрела» ячейки нумеруются от 0000 до 3777 (2047 в десятичной системе).

Каждой операции также присвоен свой номер, он записывается двузначным числом в восьмеричной системе.

Как кодируются некоторые арифметические и логические операции на машине «Стрела», показано в табл. 4.

Таблица 4

Вид операции	Код (номер) операции	Написание команды				Условия образования $\omega=1$
$+$	01	α	β	γ	01	если результат отрицателен
$-$	03	α	β	γ	03	если результат отрицателен
$\times$	05	α	β	γ	05	если абсолютная величина произведения больше или равна единице
у. п.	20	$m+x$	$m+e$	0	20	не образуется
«⟶»	13	α	0	β	13	если $(\beta)=0$
Ср.	16	α	β	0	16	если $(\alpha)\neq(\beta)$
$\oplus$	02	α	β	γ	02	не образуется
$\ominus$	15	α	β	γ	15	не образуется
стоп	40	0	0	0	40	не образуется

Кроме этих операций, в машине можно вычислять и некоторые элементарные функции, а также автоматически переводить числа из десятичной системы в двоичную и обратно по стандартным программам, постоянно хранящимся в самой машине. «Обращение» к таким программам осуществляется очень просто: например, чтобы вычислить $\sin x$, где x записан в ячейке с номером α, нужна лишь одна команда

α	0	γ	67

после выполнения которой в ячейке с номером γ будет записан $\sin x$.

В табл. 5 приведен список некоторых операций и их кодов для той же машины «Стрела», по которым вычисляются некоторые элементарные функции.

Таблица 5

Наименование операции	Код (номер) операции	Написание команды
$\sqrt{x}$	63	α 0 γ 63
e^x	64	α 0 γ 64
$\ln x$	66	α 0 γ 66
$\sin x$	67	α 0 γ 67
Перевод из десятичной системы в двоичную	72	α 0 γ 72
Перевод из двоичной системы в десятичную	70	α 0 γ 70
$\operatorname{arctg} x$	73	α 0 γ 73

В наших примерах программ первое слева число в каждой команде обозначает порядковый номер этой команды, оно указывает место команды в памяти, так как прежде чем начать работу, программа должна быть введена в запоминающее устройство машины. Эти числа также записываются в восьмеричной системе. Обычно программа занимает подряд идущие номера ячеек памяти, при этом, разумеется, безразлично, с

какой ячейки начинать нумерацию команд при составлении программы, так как все ячейки памяти машины равноправны.

Таким образом, вся программа записывается в цифровом коде.

Приведем программу нашего примера 2 для нахождения $s_n = 1 + 2 + \ldots + n$, записанную в таком цифровом коде:

0020	0027	0025	0027	01
0021	0031	0027	0031	01
0022	0026	0027	0030	03
0023	0020	0024	0000	20
0024	0000	0000	0000	40

Из этой записи видно, что каждая команда есть некоторый набор цифр в восьмеричной системе, а внутри самой машины — набор единиц и нулей (двоичная система). Распреде-

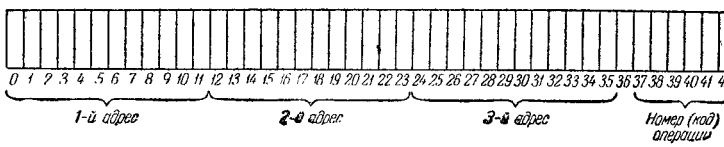


Рис. 32.

ление разрядов между адресами и кодом операции команды на разрядной сетке машины «Стрела» показано на рис. 32.

Прежде чем проследить дальнейший путь программы, обратимся к блок-схеме быстродействующей машины, изображенной на рис. 33. Такая блок-схема является в значительной мере общей для всех электронных быстродействующих машин. Жирными линиями на схеме обозначены каналы передачи числовых данных, а тонкими — каналы управления. Пунктирные линии показывают ручную передачу информации. Пульт ручного управления предназначен главным образом для целей контроля машины. Если машина исправна и программа не содержит ошибок, то оператор должен нажать единственную кнопку «начальный пуск», после чего все функции управления счетом передаются устройству автоматического управления и машина будет работать, пока не закончит решения задачи. На пульте (рис. 34) смонтировано множество переключателей, кнопок и сигнальных ламп, с помощью которых можно выполнить любую команду и проконтролировать состояние машины в любой момент времени. Можно, например, вызвать на пульт со-

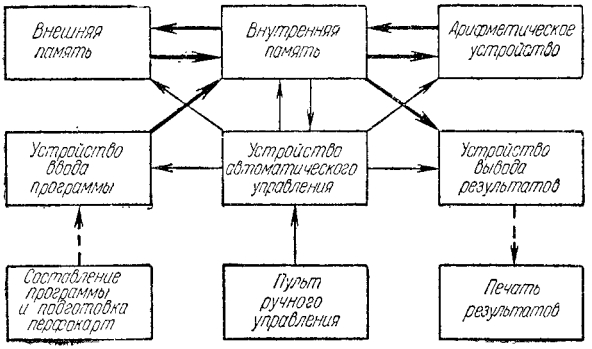


Рис. 33. Блок-схема машины.

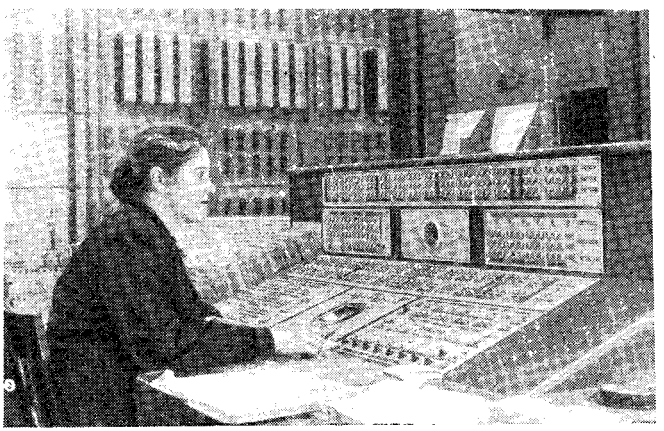


Рис. 34. Пульт машины.

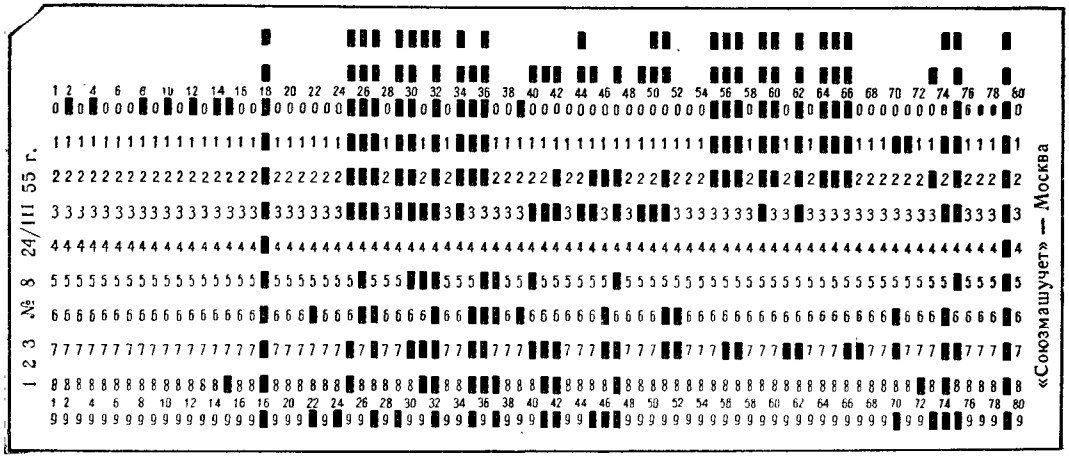


Рис. 35. Изображение перфокарты с пробитыми отверстиями.

держимое нужной ячейки памяти и убедиться в правильности хранения в ней числа, в противном случае ошибку легко исправить с пульта же.

В устройстве автоматического управления вырабатываются все необходимые импульсы для синхронизации работы блоков всей машины: осуществляется выборка из внутренней памяти очередных команд программы, расшифровка команд и выработка соответствующих импульсов управления. Наконец, устройство автоматического управления формирует и выдает на пульт сигналы, по которым оператор может судить о состоянии машины. Примерами таких сигналов могут служить: переполнение разрядной сетки (что бывает при делении на нуль), предусмотренный программой останов машины, аварийный сигнал неисправности питания и др.

Теперь вернемся к нашей программе. Записанная в восьмеричном виде программа поступает на перфоратор, где вся программа набивается на перфокарты. На рис. 35 изображена одна такая перфокарта с пробитыми числами и командами. Массив таких перфокарт поступает на читающее устройство, откуда программа вводится в запоминающее устройство машины. При вводе машина автоматически переводит всю информацию в двоичный вид и ставит первую команду на указанное программой место в памяти. Ввод перфокарт осуществляется последовательно, команды записываются в памяти также последовательно, при этом количество их указывается заранее машине. Специальный счетчик фиксирует каждую вводимую команду и таким образом определяет конец ввода. Программа введена в машину.

Затем с пульта управления нажимают кнопку «начальный пуск». Сигнал, который вырабатывается при начальном пуске, передается в устройство автоматического управления, где после нескольких вспомогательных действий осуществляется передача управления работой машины первой команде программы. Первая команда извлекается из внутренней памяти, расшифровывается и исполняется. При этом номер первой команды записывается на счетчик номера команды, расположенный в устройстве автоматического управления. Еще раз напомним, что первая команда программы в общем случае может иметь любой, но определенный для данной программы номер, и что номер команды всегда соответствует номеру ячейки памяти, где она хранится.

Расшифровка и исполнение команды происходят примерно в следующем порядке.

Устройство памяти воспринимает первый и второй адреса, отыскивает указанные ячейки и выдает их содержимое на входы всех устройств машины. Код операции также поступает на дешифраторы всех устройств, но воспринимается только арифметическим устройством. В арифметическом устройстве имеется автономная (или местная) схема управления, которая включается в действие и подготавливает при этом все необходимые импульсы для выполнения операций. По окончании операции схема местного управления арифметического устройства выдает результат для записи во внутреннюю память в ячейку,

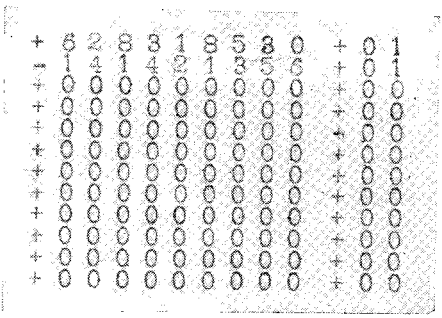


Рис. 36. Отпечатанные результаты счета.

номер которой указан в третьем адресе команды, и одновременно устройство автоматического управления прибавляет единицу в счетчик номера команды. Теперь уже на счетчике оказывается зарегистрированным номер второй команды. В начале следующего такта работы машины согласно новому номеру команды из внутренней памяти извлекается следующая команда, которая расшифровывается и исполняется подобно первой. К концу второго такта на счетчике номера команды будет записан номер третьей команды и т. д.

Таким образом, машина автоматически и последовательно будет выполнять команды программы до тех пор, пока не встретит команды условной передачи управления (у. п.). В этом случае устройство автоматического управления на предварительно очищенный счетчик номера управления запишет номер той команды, с которой нужно продолжить дальнейшие вычисления. В следующем такте машина будет исполнять именно эту команду, а затем все следующие за ней, как это было при начале работы по первой команде.

По окончании решения результаты будут выведены на устройство вывода и перфорированы на перфокарты. Массив перфокарт с результатами переносится на печатающее устройство, которое печатает результаты в нормализованной десятичной форме. Для иллюстрации два таких отпечатанных числа изображены на рис. 36.

Кроме тех команд, о которых мы кратко рассказали в этой книжке, существует еще целый ряд других, выполняющих самые разнообразные действия, как-то: передача группы чисел из внешней памяти во внутреннюю и обратно, вывод результатов из внутренней памяти на устройство вывода, арифметические операции над группами чисел и т. д. О всех таких операциях читатель может почерпнуть сведения из специальной литературы.

V. НЕАРИФМЕТИЧЕСКИЕ ПРИМЕНЕНИЯ

1. Общие замечания. Машина для игры в шахматы

Как мы уже упоминали, конструирование и применение вычислительных машин было вызвано появлением тех задач науки и техники, для решения которых, с одной стороны, требовалось произвести огромное, иногда до нескольких миллиардов, число операций, а с другой — необходимость получить это решение как можно быстрее. Приведем несколько примеров.

Прогноз погоды на завтра должен быть сделан не позже сегодняшнего дня, в противном случае такое «предсказание» не будет иметь никакой практической ценности. Между тем задача прогноза погоды сводится к решению некоторой системы дифференциальных уравнений в частных производных и требует для своего решения без применения цифровых автоматов около десяти дней. Применение машин позволило сократить время решения до нескольких часов. При изучении космических лучей требуется решать очень сложную математическую задачу, для решения которой без помощи машин потребовалось бы около 2000 лет. Машинный счет занимает всего неделю.

Таких примеров можно было бы привести очень много, например, из области ядерной физики, где встречаются особенно сложные задачи, непосильные целому коллективу математиков-вычислителей. И здесь замена «ручного» счета машинным сделала решение таких задач вполне реальным делом.

Но не только эти «арифметические» применения цифровых вычислительных машин совершили, по словам американского математика Винера, вторую промышленную революцию вслед за паром и электричеством. Наличие в таких вычислительных автоматах логических операций позволило решать и логические задачи классификации, сравнения, отбора, силлогизмов и т. п. Об этом мы и хотим коротко рассказать.

Издавна человек в своей практической деятельности применял и применяет различные механические устройства, которые сначала были очень несовершенны, но которые позволили человеку, затрачивая относительно небольшие усилия, применять физические силы, далеко превосходящие его собственные. Вся последующая эволюция техники машиностроения была направлена, по сути дела, на то, чтобы, затрачивая как можно меньше усилий, получать бы при этом наибольший эффект. Успехи промышленного прогресса современного общества подтверждают это.

Таким образом, в применении физических усилий человек значительно облегчил свой труд.

Иначе обстоит дело в области умственной деятельности. Здесь подчас исследователь, разрабатывая какую-либо научную или техническую проблему, сталкивается с рядом более мелких по своему содержанию задач, разрешение которых необходимо, но которые сводятся к чисто формальным операциям: отбор, различного рода классификация и т. п. Все это, естественно, отрывает исследователя от той важной части работы, где нужны такие факторы как интуиция, догадка и пр. Возникает вопрос: нельзя ли каким-либо образом разгрузить человеческий мозг от подобного рода работы, заставив делать это вычислительный автомат или другое устройство? И более того: возможно ли вообще построить такую машину, которая обладала бы способностью облегчать человеку его умственную деятельность в такой же мере, в какой любое механическое устройство облегчает физический труд?

Современная наука отвечает положительно на этот вопрос. И даже возможно создание такого автомата, который не будет ограничен умственными способностями его конструктора. Конечно, надо помнить, что машина может делать лишь все то из мыслительных функций человека, что может быть формализовано, т. е. записано с помощью конечного числа (пусть очень большого) конечных (подчас очень сложных) логических формул, состоящих из комбинаций трех основных логических операций: «и», «или», «нет».

Выше мы отмечали, что существуют машины-автоматы, способные моделировать многие физические процессы. Метод моделирования, сущность которого заключается в том, что поиск решения проводится не на действительном физическом объекте, а на его модели, сокращает время таких поисков. Его можно рассматривать как один из путей усиления мыслительных способностей человека. История науки знает немало

примеров, когда модель достаточно быстро подсказывала исследователю нужный путь решения. Говоря образно, именно карандаш и математические формулы, а не телескоп, помогли астрономам Леверрье и Адамсу объяснить возмущение Урана, что привело к открытию планеты Нептун. Этот расчет потребовал от них нескольких месяцев, тогда как поиск с телескопом «наугад» едва ли обеспечил бы такой успех. Если же учесть, что теперь расчет всех вариантов моделируемой системы делает с огромной скоростью вычислительный автомат, то станет ясно, что такой метод исследования таит в себе большие возможности. Путь моделирования помог осуществить, например, идею машины, способную проектировать машины, подобные себе. Проект такой машины разработан под руководством Д. Неймана.

В настоящее время ЭЦВМ могут решать очень сложные логические задачи. Среди таких «неарифметических» задач отметим две: это игра в шахматы и перевод с одного языка на другой. Мы остановимся на шахматной игре.

Существует аналогия между обучением начинающего шахматиста и «обучением» шахматной игре машины.

Первая ступень обучения человека заключается, очевидно, в том, что он должен четко усвоить правила игры, т. е. должен знать, как ходит такая-то фигура, как берет та или другая фигура и т. п. После освоения этого первого этапа наш игрок сумеет отличить одну фигуру от другой, не будет делать бессмысленных ходов, будет соблюдать очередность хода и т. д.

Добиться, чтобы машина «усвоила» правила игры в шахматы так же, как и человек на первом этапе обучения, т. е. построить соответствующую для нее программу, не представляет принципиальных трудностей, хотя такая программа будет сложна и громоздка.

Однако наш игрок отличается от опытного шахматиста тем, что не сумеет, по всей вероятности, выиграть партию у своего более опытного противника. Для этого недостаточно знания одних правил игры, нужно уметь оценивать позицию свою и противника, уметь продолжать игру, продумывая на несколько ходов вперед, и выбирать такие продолжения, которые приводят к улучшению всей позиции, а в конечном счете и к выигрышу всей партии.

Это — вторая и высшая ступень обучения игре.

При машинной игре это значит, что мы должны снабдить программу возможностью выигрывать у квалифицированного противника, иными словами, машина должна делать только

«хорошие» ходы. Эта ступень обучения машины и есть главная трудность при реализации шахматной игры на машине.

В чем же заключается эта трудность?

Доказано, что задание позиции однозначно определяет один из трех возможных исходов (выигрыш белых, ничья, выигрыш черных) при условии, что каждая сторона делает наилучшие для нее ходы. Для каждой позиции (включая и начальную) существует, очевидно, лишь конечное число возможных вариантов продолжения игры. Перебрав все возможные комбинации, мы смогли бы выбрать наилучшие ходы, однако число этих комбинаций столь велико, что даже современные счетные машины не могут выполнить эту работу. Алгорифма для точной оценки позиции существенно более простого, чем перебор всех вариантов, до сих пор не найдено (возможно, что его и нет).

Однако приблизительно («в среднем») «перевод» игры на язык формул возможен и эти средние оценки отдельных элементов игры и положены в основу анализа шахматной игры машиной.

Выше мы упомянули, что машина должна делать только «хорошие» ходы. Но что такое хороший ход? Или что такое плохой ход?

Что такое ошибка в шахматной партии? На эти вопросы разные шахматисты в разные эпохи отвечали по-разному, да и вряд ли может быть единый ответ, так как индивидуальность игры каждого квалифицированного шахматиста заключается в различном подходе к оценке позиции и в различной способности обдумывать продолжение игры на то или иное количество ходов вперед.

Тем не менее, для построения программы необходимо предусмотреть в ней возможность выбора «хорошего» хода. И здесь этот выбор производится тоже «в среднем», т. е. так как поступило бы большинство игроков в данной ситуации на доске.

Но даже и такая «усредненная» оценка хода зависит от многих причин, как-то: позиция фигур своих и противника, соотношение по качеству взятых фигур, занятость центральных полей и т. д.

При построении алгорифма этой игры в основу положена количественная оценка этих положений и количественная оценка каждой фигуры по очкам:

Король (Кр) — 200, ферзь (Ф) — 9, ладья (Л) — 5, конь (К)-слон (С) — 3, пешка (П) — 1.

Основные количественные позиционные оценки следующие: отсталая пешка ($П_1$) — 0,5, изолированная пешка ($П_2$) — 0,4, сдвоенная пешка ($П_3$) — 0,3.

Заметим, что вопрос наиболее рационального расчленения шахматной позиции на элементы и определения соответствующих количественных оценок для этих элементов представляет большую трудность и до конца не разработан.

Как и по каким законам машина оценивает позицию? В наиболее простом случае это делается так: составляется такая сумма:

$$y = 200\ (Кр - Кр') + 9\ (Ф - Ф') + 5\ (Л - Л') + 3\ (К - К' + С - С') + (П - П') + 0{,}5\ (П_1 - П_1') + 0{,}4\ (П_2 - П_2') + 0{,}3\ (П_3 - П_3') + \ldots,$$

где буквы Кр, Ф и т. д. обозначают количества соответствующих фигур у белых, а те же буквы со штрихами обозначают количества таких же фигур у черных. Машина играет белыми.

В начале игры, когда количество фигур, их позиции одинаковы у обеих сторон, $y = 0$. В процессе игры по мере подсчета машиной всех вариантов значение y меняется.

Машина оценивает улучшение позиции и успешность игры по наибольшему для данного положения значению y.

Обычно подсчет машина делает на три хода вперед. Если обозначить k-й ход белых и ответ черных соответственно через a_k и b_k, то машина сделает первый просчет своих ходов и ответов противника:

1-й вариант $a_1^{(1)}$, $b_1^{(1)}$; $a_2^{(1)}$, $b_2^{(1)}$; $a_3^{(1)}$, $b_3^{(1)}$,

и найдет значение y_1.

Просчитав 2-й вариант $a_1^{(2)}$, $b_1^{(2)}$; $a_2^{(2)}$, $b_2^{(2)}$; $a_3^{(2)}$, $b_3^{(2)}$, найдет y_2.

Вообще из n-го варианта $a_1^{(n)}$, $b_1^{(n)}$; $a_2^{(n)}$, $b_2^{(n)}$; $a_3^{(n)}$, $b_3^{(n)}$ найдется y_n.

Из вариантов, содержащих наилучшие для черных ходы, машина выберет такой вариант игры с ходами $a_1^{(l)}$, $b_1^{(l)}$; $a_2^{(l)}$, $b_2^{(l)}$; $a_3^{(l)}$, $b_3^{(l)}$, при котором y_l будет иметь наибольшее значение.

После определения такого значения y машина делает очередной ход $a_1^{(l)}$, который и будет «наилучшим» при данной «стратегии» игры.

Играя в шахматы, машина не делает «зевков», не упускает матового положения противника. Она может оказать серьезное сопротивление шахматисту среднего класса. Но шахматистов-мастеров, обдумывающих более чем на 3—4 хода вперед, машина с такой стратегией обыграть не может.

Для повышения класса машинной игры необходимо изменить такую примитивную стратегию, где принимаются во внимание даже бессмысленные ходы при просчете всех возможных вариантов, где машина «думает» за противника лишь на 3 хода вперед. Однако усиление этой стратегии в конечном счете наталкивается на еще сравнительно небольшие скорости машин (10—20 тысяч операций в секунду) и сравнительно малые объемы оперативной памяти (2—20 тысяч чисел).

Мы намеренно упрощали схему машинной игры и не излагали подробно и достаточно строго принципы программирования этой задачи, так как это заняло бы много места. Здесь мы хотели только показать, что даже такая сложная игра, как шахматы, может быть реализована в виде некоторой программы.

Интерес математиков к подобного рода задачам возник потому, что изучение теории игр применительно к машинам, разработка всех возможностей машинной игры имеют большое практическое значение.

2. Кибернетика и электронные цифровые вычислительные машины

Даже из всего сказанного понятно, что вопросы «неарифметических» применений вычислительных автоматов есть очень сложная область науки. Она оперирует с такими математическими дисциплинами, как математическая логика, теория вероятностей, теория функций, теория релейно-контактных схем и др. Иссле-

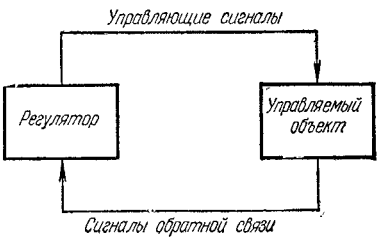


Рис. 37.

дованиями по таким применениям цифровых автоматов занимается новая наука кибернетика, возникшая около десяти лет назад и сейчас интенсивно развивающаяся. Мы попытаемся здесь очень кратко охарактеризовать ее постольку, поскольку теория автоматических быстродействующих машин составляет важную часть этого нового научного направления.

Любая автоматически управляемая система, будь то самолет-снаряд или паровая машина с регулятором скоростей, может быть охарактеризована наличием двух важнейших частей: регулятора и управляемого объекта. Эти две части в этих системах обязательно связаны между собой как прямой (от регулятора к управляемому объекту), так и обратной связью (см. схему на рис. 37). Отметим, что принципиально безразлично, соединены ли непосредственно регулятор и управляемый объект или они находятся друг от друга на значительном расстоянии. В последнем случае связь между ними может осуществляться с помощью, например, радио-сигналов.

По одному каналу (см. схему) идут сигналы, которые ставят в известность регулятор о том, что делает управляемый объект (сигналы обратной связи). Регулятор, воспринимая эти сигналы, каждый раз как бы говорит, как следует поступать управляемому объекту в том или другом положении (управляющие сигналы). Такая связь характерна и для такой сложнейшей системы управления, как нервная система животных и человека, и для такой простейшей, как, например, электрический звонок.

Главное, что нужно отметить в таких системах, какой бы сложности они ни были, это возможность переработки сигналов обратной связи в управляющие сигналы и циркуляция сигналов между регулятором и управляемым объектом.

Аналогичная связь существует и в ЭЦВМ: центральное устройство управления машины связано с арифметическим устройством прямой и обратной связью, оно «ведет» решение задачи, посылая сигналы управления как в арифметическое, так и в другие устройства.

Одним из главных вопросов, интересующих кибернетику, и является вопрос о действии различных регуляторов и создание общей теории управления. Поэтому кибернетика объединяет как математиков и техников, так физиков и биологов, ранее изучавших эти вопросы, но с различных точек зрения. Кибернетика стремится внести единые методы в их исследования.

Центральным понятием кибернетики является понятие информации. Говоря упрощенно, это — совокупность сигналов (сведения), идущих как от регулятора, так и к регулятору. Кибернетику не интересует способ передачи этих сведений, то есть соединен ли регулятор с управляемым объектом механически, с помощью радиосигналов или каким-либо другим способом. Ее не интересует и содержание этих сведений. Для нее, на-

пример, смысл сигналов: «А идет в кино» и «в В сгорел дом» безразличен. Важно лишь: 1) последовательность каких-то символов, кодирующих ту или иную информацию и 2) вероятность их появления и искажения. Последнее, например, важно при конструировании сложнейших автоматических телефонных станций, где нужна высокая надежность передачи электрических сигналов.

Электронные машины рассматриваются кибернетикой как устройства, могущие обрабатывать информацию. В самом деле, как мы убедились, поступающие исходные данные (исходная числовая информация), подвергаясь обработке по заданной наперед программе, выдаются из машины в виде результатов счета, то есть опять-таки в виде числовой информации.

Существенным элементом многих сложнейших кибернетических устройств являются именно счетные устройства. Так, в кибернетической машине, предназначенной для целей воздушной обороны, важнейшей частью является счетное устройство, которое, получая сигналы от радиолокаторов о приближении самолетов противника, вырабатывает тактический план отражения нападения в зависимости от поступивших данных, определяющих число, курс и скорость самолетов, дает сигнал воздушной тревоги, подымает самолеты в воздух, руководит воздушным боем, дает отбой и производит посадку самолетов на аэродромы.

Существует много далеко идущих аналогий между электронной вычислительной машиной и нервной системой человека. Кибернетика исследует вопросы о том, как далеко эта аналогия простирается и, насколько это доступно математическому методу, анализирует, в чем их различие.

Электронная счетная машина, как мы убедились, обладает двумя важнейшими особенностями: 1) она способна автоматически изменять ход вычислительного процесса и 2) может модифицировать команды, что позволяет преобразовывать программу в памяти машины. Эти особенности и есть, по сути дела, то главное, что отличает цифровые автоматы от любого другого счетного устройства: машина как бы сама программирует и решает задачу. Здесь имеет место некоторый самоорганизующийся процесс, который и представляет огромный интерес для кибернетики. Именно здесь находит место аналогия между работой человеческого мозга при логическом мышлении и электронной вычислительной машиной с ее самоорганизующимся процессом счета.

Известно, что каждая нервная клетка — нейрон — может находиться либо в возбужденном, либо в невозбужденном состоянии. При этом уровень возбуждения всегда одинаков и не зависит от величины раздражения. Это дает право уподоблять нейрон электронному реле, работающему по принципу «да» или «нет». Работа всей нервной системы включает в себя огромное число простейших актов («да» или «нет») отдельных клеток — нейронов. Аналогичное положение имеет место и в вычислительной машине, работающей по двоичной системе счисления.

Хотя такая схема работы нервной системы является чересчур упрощенной и схематизированной, но это и помогает выделить то главное, общее, что присуще машине и человеку.

Интересно, что существует аналогия между некоторыми заболеваниями центральной нервной системы человека и неисправностями вычислительной машины. Сущность такого заболевания нервной системы заключается в том, что в системе из большого числа нейронов возникает некий циклический процесс последовательного возбуждения нервных клеток, который может продолжаться неопределенно долго. При этом отдельные участки мозга перестают выполнять свои обычные функции. Подобный факт может иметь место и в вычислительной машине.

Мы уже отмечали, что существует свой машинный «язык» — это система команд и совокупность сигналов, реализующих условные и безусловные переходы в программе. Этот «язык» позволяет воспроизводить в машине некоторые логические процессы, подобные мышлению человека.

Огромную роль должно сыграть применение электронных машин для моделирования процессов высшей нервной деятельности. И если раньше физиологи могли лишь наблюдать за работой мозга, то теперь они могут, применяя модели, воспроизводить некоторые, правда, пока очень примитивные, процессы логического мышления.

Мы не будем приводить примеры таких моделей, так как их следует рассматривать лишь как первые шаги в деле познания наиболее сложного органа — центральной нервной системы человека.

Но не следует, однако, преувеличивать все возможности вычислительных машин. Создатель современной кибернетики Н. Винер подчеркивает, что мыслительные способности машины определяются не ее свойствами и конструктивными особенностями, а целиком зависят от той программы, которая составляется и вводится в машину человеком. Любая счетная машина

без программы, заложенной в нее человеком,— это груда металла, не способная ни к каким самостоятельным действиям. Это нужно всегда иметь в виду, когда проводится слишком тесная аналогия между машиной и ее творцом — человеком.

Заканчивая на этом наш короткий рассказ о кибернетике, скажем, что ее теоретические выводы уже принесли и, несомненно, принесут еще громадную пользу в науке, промышленности и в повседневной жизни. Мы не смогли в этой главе рассказать о многих интересных и важных задачах, стоящих перед кибернетикой, как, например, о переводе с одного языка на другой, управлении производственными процессами, о моделировании зрения и речи, об изучении биотоков. Читатель может найти ответы на эти вопросы в специальных книгах и журналах.

ПРИЛОЖЕНИЕ

В заключение приведем краткие характеристики некоторых отечественных и зарубежных автоматических цифровых вычислительных машин.

1. БЭСМ Академии наук СССР

Универсальная машина параллельного действия с плавающей запятой, выполняющая 8—10 тыс. операций в секунду. Порядок p нормализованного числа для этой машины заключен в пределах: $-31 \leqslant p \leqslant 31$. Система команд трехадресная. Внутренняя память машины двух видов: собственно внутреннее запоминающее устройство (ВЗУ) и диодное запоминающее устройство (ДЗУ). ВЗУ выполнено на ферритовых сердечниках и имеет емкость 1023 ячейки. ДЗУ имеет емкость 384 ячейки и является «пассивной» памятью: его содержимое не может изменяться во время работы машины. Оно предназначено для хранения некоторых констант и стандартных подпрограмм, которые могут быть набраны ручным способом. Внешняя память двух видов: магнитный барабан на 5120 чисел и четыре магнитные ленты общей емкостью 120 тыс. чисел. В отличие от большинства вычислительных машин, на которых для выборки команд из внутренней памяти применяется счетчик команд, БЭСМ имеет два таких счетчика и две связанные с ними системы управления — центральную и местную. Программа решения задачи может быть составлена для работы на любой системе управления, но обычно программы составляются для центральной системы управления, а местное управление используется при обращении к подпрограммам. Для команд передачи управления имеются специальные команды, переключающие машину с одной системы управления на другую.

Для изображения числа (или команды) отведено 39 двоичных разрядов, причем порядок занимает пять двоичных разрядов ($(11111)_2=31$) и один разряд для знака порядка, а мантисса числа занимает 32 двоичных разряда и один разряд для знака числа.

Трехадресная команда БЭСМ имеет вид:

код операции	α	β	γ

где α, β и γ — первый, второй и третий адреса команды. Распределение разрядов между адресами и кодом операции следующее: код операции занимает первые шесть разрядов, на каждый адрес отведено 11 разрядов.

Машина имеет фотоэлектрическое устройство ввода в ВЗУ с перфоленты. Выводное устройство двух видов: электромеханическая печать (1,5 числа в секунду) и фотопечать для выдачи больших массивов информации (200 чисел в секунду).

2. «Стрела» (СССР)

Универсальная машина параллельного действия с плавающей запятой. Система команд трехадресная. Для изображения чисел в машине отведено 43 разряда. Внутренняя память на электронно-лучевых трубках емкостью 2048 чисел. Внешняя память на магнитных лентах общей емкостью до 500 тыс. чисел. Есть еще дополнительное запоминающее устройство емкостью 256 ячеек для постоянного хранения некоторых наиболее часто встречающихся констант (π, e, $\sqrt{2}$ и др.) и стандартных подпрограмм (вычисление некоторых элементарных функций, перевод чисел из десятичной системы в двоичную и обратно и др.). Стандартные подпрограммы в дополнительном запоминающем устройстве набираются ручным способом. Средняя скорость работы машины около двух тысяч операций в секунду. Ввод и вывод информации на перфокартах.

3. М-2 (СССР)

Универсальная машина Энергетического института Академии наук СССР. Арифметическое устройство параллельного действия. Машина способна работать в двух режимах: плавающая и фиксированная запятая. Система команд трехадресная. Для

изображения чисел в машине отведено 34 разряда. Внутренняя память двух видов: основная — на электронно-лучевых трубках емкостью в 512 чисел, дополнительная — на магнитном барабане. Внешняя память — на магнитной ленте на 50 тыс. чисел. Скорость работы машины около 2 тыс. операций в секунду. Устройств ввода три: основное — фотоэлектрическое (около 1500 двоичных знаков в секунду); дополнительное вводное устройство, работающее со значительно меньшей скоростью и предназначенное для записи непосредственно на барабан; вспомогательное — клавишное для внесения небольших исправлений в исходные данные. Выводное устройство печатает результаты в виде таблиц и одновременно перфорирует ленту. Следует отметить, что машина может работать с «*двойной точностью*», т. е. для записи одного числа используются две ячейки (число разрядов увеличивается в два раза).

4. «Урал» (СССР)

Универсальная автоматическая цифровая машина параллельно-последовательного действия с фиксированной запятой. Система команд одноадресная. Для изображения чисел в машине отведено 36 разрядов. Внутренняя память — магнитный барабан на 1023 числа (на 2047 команд). Внешняя память — на магнитной ленте (на 40 тыс. чисел или 80 тыс. команд). Машина может выполнять 29 различных операций. Скорость работы — около 100 операций в секунду.

5. СЦМ (СССР)

Специализированная цифровая машина (СЦМ) предназначена для решения систем обыкновенных дифференциальных уравнений и других задач, решение которых сводится к повторению определенной последовательности арифметических операций. Одноадресная машина последовательного действия с фиксированной запятой. Для изображения чисел отведено 32 двоичных разряда. Запоминающее устройство — магнитный барабан емкостью 1856 команд и 124 числа. Ввод — автоматический с перфоленты и ручной с пульта. Результаты выводятся в десятичной форме и печатаются электрифицированными пишущими машинами.

6. САПО (Чехословакия)

Быстродействующая вычислительная машина параллельного типа с плавающей запятой. Система команд пятиадресная: если результат операции положителен, то машина переходит к команде, порядковый номер которой записан в четвертом адресе, в противном случае, если результат отрицателен, машина переходит к выполнению команды, номер которой указан в пятом адресе. Запоминающее устройство — магнитный барабан емкостью 1024 числа. Для изображения числа отведено 32 двоичных разряда. Правильность вычисленных результатов обеспечивается тройным универсальным арифметическим устройством и автоматическим обнаружением, регистрацией и исправлением ошибок.

7. ИБМ-702 фирмы ИБМ (США)

Универсальная автоматическая цифровая машина последовательного действия, одноадресная. Внутреннее запоминающее устройство на специальных электростатических трубках конструкции той же фирмы емкостью 10 тыс. чисел. Внешняя память — на магнитном барабане емкостью 60 тыс. чисел. «Длина» кодов (количество двоичных разрядов в числе или команде) не является постоянным для данной машины и может быть изменена по желанию программиста. Машина выполняет 32 различные операции со скоростью около 4 тыс. сложений, или около 800 умножений в секунду. Ввод и вывод — на перфокартах со скоростью 250 и 100 перфокарт в минуту соответственно. В машине предусмотрен контроль за правильностью хранения информации.

8. «Ураган» фирмы Райтеон (США)

Универсальная вычислительная машина последовательного действия с фиксированной запятой. Система команд четырехадресная. Для изображения чисел отведено 30 разрядов. Внутренняя память — на ртутных линиях задержки емкостью 1152 числа. Внешняя память — на четырех магнитных лентах емкостью по 10 тыс. чисел каждая. Машина выполняет операции сложения за 48 *мксек*, умножения за 285 *мксек* и деления за 430 *мксек*. Имеется система контроля, останавливающая машину при ошибках в передаче информации или при выполнении арифметических операций.

9. «Диджитак» (США)

Самолетная цифровая вычислительная машина для автоматического управления самолетом и бомбометания. Двухадресная машина последовательного действия с фиксированной запятой, оперирующая с 17-разрядными двоичными числами. Запоминающее устройство — на магнитном барабане. Скорость вычислений (около 700 операций в секунду) может быть увеличена за счет того, что результат предыдущей операции не засылается в запоминающее устройство, а является исходным для следующей операции. В машину введена специальная программа, контролирующая входные данные (например, с радиолокатора) от случайных ошибок. Размеры машины небольшие: 500×650×475 *мм*. Вес собственно машины — 52 *кг*.

10. LGP-30 фирмы Лайбраскоп (США)

Универсальная вычислительная одноадресная машина с фиксированной запятой. Для изображения чисел в машине отведено 32 разряда. Запоминающее устройство — магнитный барабан емкостью 4096 чисел. Скорость работы — около 100 операций в секунду. Для целей контроля предусмотрена возможность шаговой работы.

11. УНИВАК I фирмы Эккерт — Моучли (США)

Универсальная машина с запоминающим устройством на ртутных линиях задержки (емкость 91 тыс. чисел) и магнитная лента (10 катушек общей емкостью около 15 млн. двоичных чисел). Машина производит операцию сложения за 600 *мксек*, умножения — за 2500 *мксек*, деления — за 4000 *мксек*. Ввод и вывод — на магнитных лентах. В машине есть устройство контроля, способное обнаруживать ошибку и указывать ее характер.

12. УНИВАК II той же фирмы

Модернизированный вариант предыдущей машины. Запоминающее устройство — на магнитных сердечниках емкостью 24 тыс. двоичных чисел. Емкость может быть увеличена до 120 тысяч двоичных чисел. Ввод и вывод — на магнитной ленте со скоростью 20 тыс. знаков в 1 сек. Машина применяется для коммерческих расчетов.

13. «Пегас» фирмы Ферранти (Англия)

Универсальная автоматическая цифровая машина средних размеров, последовательного действия, с фиксированной запятой, одноадресная. Машина оперирует с 39-разрядными двоичными числами. Внутренняя память — на никелевых линиях задержки. Внешняя память — на магнитном барабане емкостью 4096 чисел. Машина выполняет операцию сложения за 2 *мксек* и деления за 5 *мксек*. Ввод и вывод может производиться как на перфокартах, так и на магнитных лентах.

14. «Меркурий» той же фирмы

Универсальная машина последовательно-параллельного действия с плавающей запятой. Внутренняя память — на ферритах, емкостью 1024 числа, имеется дополнительная память на четырех магнитных барабанах (по 16 тыс. чисел каждый). Машина выполняет операцию сложения за 60—180 *мксек* и умножения за 300 *мксек*. Ввод информации осуществляется на перфоленте со скоростью 400 знаков в одну секунду.

15. АПЕ(КС)К (Англия)

Универсальная машина Лондонского университета последовательного типа, двухадресная. Емкость запоминающего устройства 512 двоичных чисел. Сложение и вычитание происходит за 600 *мксек*, умножение — за 18 *мксек*. Машина, помимо учебных целей, предназначена для исследовательской работы.

16. ЭДСАК (Англия)

Универсальная вычислительная машина последовательного типа с фиксированной запятой, одноадресная. Для изображения чисел отведено 17 двоичных разрядов. Возможен режим «двойной точности», когда число изображается 34 двоичными разрядами. Память на ультразвуковых линиях задержки емкостью 1024 двоичных числа. Машина выполняет 1—2 тыс. операций в секунду. Ввод и вывод на перфоленте.

17. ЛЕО фирмы Лайонс (Англия)

Машина построена специальным отделом этой фирмы, которая владеет большим количеством кафе, столовых и предприятиями, выпускающими кондитерские изделия. Машина предназначена для

коммерческих целей, как-то: начисление заработной платы рабочим и служащим, планирование производства и т. п. Выдача результатов счета происходит непосредственно на карточках и бланках, используемых далее в документационной работе фирмы. Машина снабжена разнообразными вводными и выводными устройствами, обеспечивающими ввод и вывод большого количества данных, необходимых при коммерческих расчетах.

18. ПЭРМ (ФРГ)

Универсальная машина Мюнхенской высшей технической школы (ФРГ). Параллельного типа, одноадресная, способна работать в двух режимах: плавающая и фиксированная запятая. Для изображения чисел отведено 50 двоичных разрядов. Запоминающее устройство — на магнитном барабане емкостью около 8 тыс. чисел. Скорость работы — около 250 операций в секунду. Ввод информации осуществляется с перфокарт или с магнитной ленты. Машина может работать автоматически в шаговом режиме.

ОГЛАВЛЕНИЕ

От авторов 3
Введение 5

I. Общие сведения о вычислительных машинах

1. Типы вычислительных машин. Области применения 7
2. Основные устройства универсальных и специализированных цифровых машин и их характеристики 19

II. Арифметические основы действия вычислительных машин

1. Изображение чисел с помощью письменных знаков. Системы счисления 22
2. Нормальная форма представления чисел. Плавающая и фиксированная запятая 29
3. Об изображении отрицательных чисел 33
4. Арифметические действия, совершаемые машиной 34

III. Элементы электронной цифровой вычислительной машины

1. Логические элементы 43
2. Арифметическое устройство 51
3. Запоминающее устройство 56
4. Устройства ввода и вывода 77
5. Контроль исправности 80

IV. Программирование и решение задач

1. Команды 84
2. Программирование задач для цифровых вычислительных машин 88
3. Подготовка задачи к программированию и решение ее на цифровой вычислительной машине 101

V. Неарифметические применения

1. Общие замечания. Машина для игры в шахматы 110
2. Кибернетика и электронные цифровые вычислительные машины 115

Приложение 120

Архангельский Николай Алексеевич
и *Зайцев Борис Ильич.*

Автоматические цифровые машины.

Редактор *А. А. Коноплянкин.*

Техн. редактор *Е. А. Ермакова.*
Корректор *Г. Г. Желтова.*

Сдано в набор 30/VII 1958 г. Подписано к печати 17/XI 1958 г. Бумага 84×108 1/32. Физ. печ. л. 4,0. Условн. печ. л. 6,56. Уч.-изд. л. 5,83. Т-11524. Тираж 50 000 экз. Цена книги 1 р. 75 к. Заказ 2131.

Государственное издательство физико-математической литературы. Москва, В-71, Ленинский проспект, 15.

Первая Образцовая типография имени А. А. Жданова Московского городского Совнархоза. Москва, Ж-54, Валовая, 28.

сравниваются. Если результаты двух просчетов совпали, то машина сама переходит к решению другой части задачи.

Одним из эффективных методов проверки хранения и передачи информации является метод контрольного суммирования. Метод заключается в том, что вся информация, подлежащая хранению или передаче в другое устройство, независимо от того, чи́сла это или команды, суммируется и сумма запоминается. При проверке хранения или при передаче информации производится повторное ее суммирование. Машина автоматически сравнивает первую и вторую сумму и в случае их несовпадения останавливается.

Быстрое нахождение неисправностей требует хорошо подготовленной документации и детального знания оператором всех узлов машины. Различные диаграммы временных соотношений импульсов, карты напряжений важнейших точек схемы, измерительные приборы и т. д. способствуют быстрому устранению неисправностей.

Заманчивыми и реальными являются способы автоматического обнаружения и исправления случайных ошибок вычислений, на которых мы не имеем возможности останавливаться.

Особенно следует отметить то, что вычислительная машина состоит из большого количества взаимозаменяемых ячеек. Это существенно облегчает устранение неисправностей и сводит такую работу к замене дефектной ячейки на заведомо хорошую. На рис. 30 приведена фотография нескольких сменных ячеек арифметического устройства машины «Стрела». Сменные ячейки вставляются на стойки (или панели) с помощью контактных разъемов. Обратные, или монтажные стороны панелей несут на себе чрезвычайно большое количество соединительных проводов. На рис. 31 приведена фотография монтажной стороны одной из стоек машины «Стрела». Нетрудно представить, сколько нужно знаний и опыта, чтобы быстро ориентироваться в столь сложной паутине электрических нервов, молниеносно и в бесчисленных комбинациях соединяющих между собой десятки и сотни тысяч различных радиодеталей.